```
1000  .PAGE "OSS SYSTEM EQUATES FOR ATARI"
1010 ;
1020 ;  FILE = #DN:SYSEQU.ASM
1030 ;
1040 ;
1050 ; I/O CONTROL BLOCK EQUATES
1060 ;
1070 IOCB *= $0340  ;START OF SYSTEM IOCBS
1080 ;
1090 ICHID *= *+1 ;DEVICE HANDLER IS (SET BY OS)
1100 ICDNO *= *+1 ;DEVICE NUMBER (SET BY OS)
1110 ICCOM *= *+1 ;I/O COMMAND
1120 ICSTA *= *+1 ;I/O STATUS
1130 ICBADR *= *+2 ;BUFFER ADDRESS
1140 ICPUT *= *+2 ;DH PUT ROUTINE (ADR-1)
1150 ICBLEN *= *+2 ;BUFFER LENGTH
1160 ICAUX1 *= *+1 ;AUX 1
1170 ICAUX2 *= *+1 ;AUX 2
1180 ICAUX3 *= *+1 ;AUX 3
1190 ICAUX4 *= *+1 ;AUX 4
1200 ICAUX5 *= *+1 ;AUX 5
1210 ICAUX6 *= *+1 ;AUX 6
1220 ;
1230 IOCBLEN = *-IOCB ;LENGTH OF ONE IOCB
1240 ;
1250 ; IOCB COMMAND VALUE EQUATES
1260 ;
1270 COPN = 3 ;OPEN
1280 CGBINR = 7 ;GET BINARY RECORD
1290 CGTXTR = 5 ;GET TEXT RECORD
1300 CPBINR = 11 ;PUT BINARY RECORD
1310 CPTXTR = 9 ;PUT TEXT RECORD
1320 CCLOSE = 12 ;CLOSE
1330 CSTAT = 13 ;GET STATUS
1340 ;
1350 ; DEVICE DEPENDENT COMMAND EQUATES FOR FILE MANAGER
1360 ;
1370 CREN = 32 ;RENAME
1380 CERA = 33 ;ERASE
1390 CPRO = 35 ;PROTECT
1400 CUNP = 36 ;UNPROTECT
1410 CPOINT = 37 ;POINT
1420 CNOTE = 38 ;NOTE
1430 ;
1440 ; AUX1 VALUES REQD FOR OPEN
1450 ;
1460 OPIN = 4 ;OPEN INPUT
1470 OPOUT = 8 ;OPEN OUTPUT
1480 OPUPD = 12 ;OPEN UPDATE
1490 OPAPND = 9 ;OPEN APPEND
1500 OPDIR = 6 ;OPEN DIRECTORY
1510 ;
1520  .PAGE
1530 ;
1540 ;    EXECUTE FLAG DEFINES
1550 ;
1560 EXCYES = $80 ; EXECUTE IN PROGRESS
1570 EXCSCR = $40 ; ECHO EXCUTE INPUT TO SCREEN
1580 EXCNEW = $10 ; EXECUTE START UP MODE
1590 EXCSUP = $20 ; COLD START EXEC FLAG
1600 ;
1610 ; MISC ADDRESS EQUATES
1620 ;
1630 CPALOC = $0A ;.POINTER TO CP/A
```

```
1640 WARMST = $08 ; WAR, START (0=COLD)
1650 MEMLO = $2E7 ; AVAIL MEM (LOW) PTR
1660 MEMTOP = $2E5 ; AVAIL MEM (HIGH) PTR
1670 APPMHI = $0E ; UPPER LIMIT OF APPLICATION MEMORY
1680 INITADR = $2E2 ; ATARI LOAD/INIT ADR
1690 GOADR = $2E0 ; ATARI LOAD/GO ADR
1700 CARTLOC = $BFFA ; CARTRIDGE RUN LOCATION
1710 CIO = $E456 ;CIO ENTRY ADR
1720 EOL = $9B ; END OF LINE CHAR
1730 ;
1740 ;  CP/A FUNCTION AND VALUE DISPLACEMENT
1750 ;      (INDIRECT THROUGH CPALOC)
1760 ;              IE. (CPALOC),Y
1770 ;
1780 CPGNFN = 3 ; GET NEXT FILE NAME
1790 CPDFDV = $07 ; DEFAULT DRIVE (3 BYTES)
1800 CPBUFP = $0A ; CMD BUFF NEXT CHAR POINTR (1 BYTE)
1810 CPEXFL = $0B ; EXECUTE FLAG
1820 CPEXFN = $0C ; EXECUTE FILE NAME (16 BYTES)
1830 CPEXNP = $1C ; EXECUTE NOTE/POINT VALUES
1840 CPFNAM = $21 ; FILENAME BUFFER
1850 RUNLOC = $3D ; CP/A LOAD/RUN ADR
1860 CPCMDB = $40 ; COMMAND BUFFER (60 BYTES)
```

Exhibit I

Sample, Reproducible
ATARI Programming Form

PROGRAM	PAGE	DATE
LOOPNEST.SRC	1 OF 1	4/18/82
this program works	PROGRAMMER E.B.	

LINE NO.	LABEL	OP CODE	OPERAND	COMMENT
10		*=	$2100	
20		CLC		
30		LDA	#$0	
40	OUTLP	ADC	#$01	
50		CMP	#$02	
60		BNE	INLOOP	
70		JMP	END	
80	INLOOP	PHP		
90		LDX	#$00	
100	UP	INX		
110		CPX	#$02	
120		BNE	UP	
130		PLP		
140		JMP	OUTLP	
150	END			

--- general remarks ---

```
                0210 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                0310 ;
                0410 ; The "M:" driver --
                0510 ;    Using memory as a device
                0610 ;
                0710 ; Includes installation program
                0810 ;
                0910 ; Written by Bill Wilkinson
                1010 ;   for January, 1982, COMPUTE!
                1110 ;
                1210 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                1310 ; ASM ,#P:,#D:MEM.OBJ
                1410 ; SAVE#D:MEM.ASM
                1510 ;
                1610 ; EQUATES INTO ATARI'S OS, ETC.
                1710 ;
=034A           1810 ICAUX1 = $034A  ; The AUX1 byte of IOCB
                1910 ;
=0008           2010 OPOUT = 8       ; Mode 8 is OPEN for OUTPUT
                2110 ;
=02E7           2210 MEMLO = $02E7   ; pointer to bottom of free RAM
=02E5           2310 MEMTOP = $02E5  ; pointer to top of free RAM
                2410 ;
=00E0           2510 FR1 =   $E0     ; Fltg Pt Register 1, scratch
                2610 ;
=0001           2710 STATUSOK = 1    ; I/O was good
=0088           2810 STATUSEOF = $88 ; reached an end-of-file
                2910 ;
=031A           3010 HATABS = $031A
                3110 ;
=0100           3210 HIGH =  $0100   ; divisor for high byte
=00FF           3310 LOW =   $FF     ; mask for low byte
                3410 ;
                3510     .PAGE "The installation routine"
```

```
                    3610 ;======= CHANGE NEXT LINE TO SUIT YOUR MEMORY =======
0000                3710     *=  $3000
                    3810 ; This first routine is simply
                    3910 ; used to connect the driver
                    4010 ; to Atari's handler address
                    4110 ; table.
                    4210 ;
3000                4310 LOADANDGO
3000 A200           4410     LDX #0       ; We begin at start of table
3002                4510 SEARCHING
3002 BD1A03         4610     LDA HATABS,X ; Check device name
3005 F00A           4710     BEQ EMPTYFOUND ; Found last one
3007 C94D           4810     CMP #'M      ' ; Already have M: ?
3009 F01A           4910     BEQ MINSTALLED ; Yes, don't reinstall
300B E8             5010     INX
300C E8             5110     INX
300D E8             5210     INX          ; Point to next entry
300E D0F2           5310     BNE SEARCHING ; and keep looking
3010 60             5410     RTS          ; Huh? Impossible!!!
                    5510 ;
                    5610 ; We found the current end of the
                    5710 ; table...so extend it.
                    5810 ;
3011                5910 EMPTYFOUND
3011 A94D           6010     LDA #'M      ' ; Our device name, "M:"
3013 9D1A03         6110     STA HATABS,X ; is first byte of entry
3016 A93B           6210     LDA #MDRIVER&LOW
3018 9D1B03         6310     STA HATABS+1,X ; LSB of driver addr
301B A930           6410     LDA #MDRIVER/HIGH
301D 9D1C03         6510     STA HATABS+2,X ; and MSB of addr
3020 A900           6610     LDA #0
3022 9D1D03         6710     STA HATABS+3,X ; A new end for the table
                    6810 ;
                    6910 ; now change LOMEM so BASIC won't
                    7010 ; overwrite us.
                    7110 ;
3025                7210 MINSTALLED
3025 A900           7310     LDA #DRIVERTOP&LOW
3027 8DE702         7410     STA MEMLO   ; LSB of top addr
302A A931           7510     LDA #DRIVERTOP/HIGH
302C 8DE802         7610     STA MEMLO+1 ; and MSB therof
                    7710 ;
                    7810 ; and that's all we have to do!
                    7910 ;
302F 60             8010     RTS
                    8110 ;
                    8210 ;
                    8310 ;;;;;;;;;;;;;;;;;;;;;;;;;;;
                    8410 ;
                    8510 ; This entry point is provided
                    8610 ; so that BASIC can reconnect
                    8710 ; the driver via a USR(RECONNECT)
                    8810 ;
3030                8910 RECONNECT
3030 68             9010     PLA
3031 F0CD           9110     BEQ LOADANDGO ; No parameters, I hope
3033 A8             9210     TAY
3034                9310 PULLTHEM
3034 68             9410     PLA
3035 68             9510     PLA          ; get rid of a parameter
3036 88             9610     DEY
```

```
3037 D0FB       9710     BNE PULLTHEM ; and pull another
3039 F0C5       9810     BEQ LOADANDGO; go reconnect
                9910 ;
                010010   .PAGE "The driver itself"
```

```
                    010110 ;
                    010210 ; Recall that all drivers must
                    010310 ; be connected to OS through
                    010410 ; a driver routines address table.
                    010510 ;
303B                010610 MDRIVER
303B 4C30           010710   .WORD MOPEN-1 ; The addresses must
303D 6F30           010810   .WORD MCLOSE-1 ; ...be given in this
303F 9230           010910   .WORD MGETB-1 ; ...order and must
3041 8530           011010   .WORD MPUTB-1 ; ...be one (1) less
3043 9F30           011110   .WORD MSTATUS-1 ; ...than the actual
3045 4930           011210   .WORD MXIO-1 ; ...address
3047 4C4A30         011310   JMP MINIT   ; This is for safety only
                    011410 ;
                    011510 ; For many drivers, some of these
                    011610 ; routines are not needed, and
                    011710 ; can effectively be null routines
                    011810 ;
                    011910 ; A null routine should return
                    012010 ; a one (1) in the Y-register
                    012110 ; to indicate success.
                    012210 ;
304A                012310 MXIO
304A                012410 MINIT
304A A001           012510   LDY #1      ; success
304C 60             012610   RTS
                    012710 ;
                    012810 ; If a routine is omitted because
                    012910 ; it is illegal (reading from a
                    013010 ; printer, etc.), simply pointing
                    013110 ; to an RTS is adequate, since
                    013210 ; Atari OS preloads Y with a
                    013310 ; 'Function Not Implemented' error
                    013410 ; return code.
                    013510 ;
                    013610   .PAGE "The driver function routines"
```

```
                    013710 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                    013810 ;
                    013910 ; Now we begin the code for the
                    014010 ; routines that do the actual
                    014110 ; work
                    014210 ;
304D                014310 MOPEN
304D BD4A03         014410   LDA ICAUX1,X ; Check type of open
3050 2908           014510   AND #OPOUT   ; Open for output?
3052 F00D           014610   BEQ OPENFORREAD ; No...assume for input
3054 ADE502         014710   LDA MEMTOP
3057 8DD230         014810   STA MSTART   ; We start storing
305A ACE602         014910   LDY MEMTOP+1 ; ...the bytes
305D 88             015010   DEY          ; ...one page below
305E 8CD330         015110   STY MSTART+1 ; the supposed top of mem
                    015210 ;
                    015310 ; now we join up with mode 4 open
                    015410 ;
3061                015510 OPENFORREAD
3061 ADD230         015610   LDA MSTART   ; simply move the
3064 8DCE30         015710   STA MCURRENT ; ...start pointer
3067 ADD330         015810   LDA MSTART+1 ; ...to the current
306A 8DCF30         015910   STA MCURRENT+1 ; ...pointer, both bytes
                    016010 ;
306D A001           016110   LDY #STATUSOK
306F 60             016210   RTS          ; we don't acknowledge failure
                    016310 ;
                    016410 ;
                    016510 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                    016610 ;
                    016710 ; the routine for CLOSE of M:
                    016810 ;
3070                016910 MCLOSE
3070 BD4A03         017010   LDA ICAUX1,X ; check mode of open
3073 2908           017110   AND #OPOUT   ; was for output?
3075 F00C           017210   BEQ MCLREAD ; no...close input 'file'
                    017310 ;
3077 ADCE30         017410   LDA MCURRENT ; we establish our
307A 8DD030         017510   STA MSTOP    ; ...limit so that
307D ADCF30         017610   LDA MCURRENT+1 ; ...next use can't
3080 8DD130         017710   STA MSTOP+1 ; ...go too far
                    017810 ;
3083                017910 MCLREAD
3083 A001           018010   LDY #STATUSOK
3085 60             018110   RTS          ; and guaranteed to be ok
                    018210 ;
                    018310 ;
                    018410 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                    018510 ;
                    018610 ; This routine puts one byte
                    018710 ; to the memory for later
                    018810 ; retrieval.
                    018910 ;
3086                019010 MPUTB
3086 48             019110   PHA          ; save the byte to be PUT
3087 20B530         019210   JSR MOVECURRENT ; get ptr to zero page
308A 68             019310   PLA          ; the byte again
308B A000           019410   LDY #0
308D 91E0           019510   STA (FR1),Y ; put the byte, indirectly
308F 20C030         019610   JSR DECCURRENT ; point to nxt byte
3092 60             019710   RTS          ; that's all
```

```
                    019810 ;
                    019910 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                    020010 ;
                    020110 ; routine to get a byte put
                    020210 ; in memory before.
                    020310 ;
3093                020410 MGETB
3093 20A030         020510   JSR MSTATUS ; any more bytes?
3096 B007           020610   BCS MGETRTS ; no...error
3098 A000           020710   LDY #0
309A B1E0           020810   LDA (FR1),Y ; yes...get a byte
309C 20C030         020910   JSR DECCURRENT ; and point to next byte
309F                021010 MGETRTS
309F 60             021110   RTS
                    021210 ;
                    021310 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                    021410 ;
                    021510 ; check the status of the driver
                    021610 ;
                    021710 ; this routine is only valid
                    021810 ; when READing the 'file'...
                    021910 ; "M:" never gets errors when
                    022010 ; writing.
                    022110 ;
30A0                022210 MSTATUS
30A0 20B530         022310   JSR MOVECURRENT ; current ptr to zero page
30A3 CDD030         022410   CMP MSTOP   ; any more bytes to get?
30A6 D009           022510   BNE MSTOK   ; yes
30A8 CCD130         022610   CPY MSTOP+1 ; double chk
30AB D004           022710   BNE MSTOK   ; yes, again
30AD A088           022810   LDY #STATUSEOF ; oops...
30AF 38             022910   SEC         ; no more bytes
30B0 60             023010   RTS
                    023110 ;
30B1                023210 MSTOK
30B1 A001           023310   LDY #STATUSOK ; all is okay
30B3 18             023410   CLC         ; flag for MGETB
30B4 60             023510   RTS
                    023610   .PAGE "Miscellaneous subroutines"
```

```
                    023710 ;
                    023810 ;;;
```

```
                    027910 ;
                    028010 ;;;
```

```
7PAGE 1   ***** KATHY'S COLOR PALLETTE *****

                    10       .TITLE "***** KATHY'S COLOR PALLETTE *****"
                    20 ;PUTS ALL 128 COLORS ONTO THE SCREEN
                    30 ;FROM ATARI ASSEMBLER/EDITOR REFERENCE MANUAL
                    40 ;PP.72-74
                    50 ;CALL FROM BASIC WITH THE FOLLOWING COMMANDS:
                    60 ; ASM ,#P:,#D:KATY.OBJ
                    70 ;GRAPHICS 5
                    80 ;FOR I=0 TO 3:COLOR I:FOR J=20*I TO 20*I+19:PLOT J,3
                    90 ;DRAWTO J,39:NEXT J:NEXT I
                    0100 ;A=USR(1664)
                    0110 ;
                    0120 ;BASIC IS STILL USABLE
                    0130 ;EXIT WITH SYSTEM RESET KEY
                    0140 ;
0000                0150     *=  $0680
    =00CC           0160 POINTA = $CC     POINTER TO DISPLAY LIST
    =00CE           0170 COLCNT = $CE     KEEPS TRACK OF COLOR WE ARE ON
    =00CF           0180 DECK =  $CF      BIT 0 KEEPS TRACK OF WHICH DECK
    =0230           0190 DSLSTL = $0230   O.S. DISPLAY LIST ADDRESS
    =D40E           0200 NMIEN = $D40E    NON-MASKABLE INTERRUPT ENABLE
    =D40F           0210 NMIRES = $D40F   NON-MASKABLE INTERRUPT RESET
    =D40F           0220 NMIST = $D40F    NON-MASKABLE INTERRUPT STATUS
    =0200           0230 VDSLST = $0200   DISPLAY LIST INTERRUPT VECTOR
    =D01A           0240 COLBAK = $D01A   BACKGROUND COLOR REGISTER
    =D016           0250 COLPF0 = $D016   COLOR REGISTER #0
    =D017           0260 COLPF1 = $D017   COLOR REGISTER #1
    =D018           0270 COLPF2 = $D018   COLOR REGISTER #2
    =D40A           0280 WSYNC = $D40A    WAIT FOR HORIZONTAL SYNC
0680 68             0290 SETUP PLA        CLEAN STACK
                    0300 ;SET UP POINTER ON PAGE ZERO
                    0310 ;
0681 AD3002         0320     LDA DSLSTL
0684 85CC           0330     STA POINTA
0686 AD3102         0340     LDA DSLSTL+1
0689 85CD           0350     STA POINTA+1
                    0360 ;
068B A007           0370     LDY #$07     POINT TO 3RD MODE BYTE
068D A98A           0380     LDA #$8A     NEW MODE BYTE
                    0390 ;
                    0400 ;STORE 16 DISPLAY LIST INTERRUPT MODE BYTES
                    0410 ;
068F 91CC           0420 LOOP1 STA (POINTA),Y
0691 C8             0430     INY
0692 C017           0440     CPY #$17
0694 D0F9           0450     BNE LOOP1
                    0460 ;
                    0470 ;SKIP FOUR BLANK LINES
                    0480 ;
0696 C8             0490     INY
0697 C8             0500     INY
0698 C8             0510     INY
0699 C8             0520     INY
                    0530 ;
                    0540 ;STORE 16 MORE DL INTERRUPT MODE BYTES
                    0550 ;
069A 91CC           0560 LOOP2 STA (POINTA),Y
069C C8             0570     INY
069D C02B           0580     CPY #$2B
069F D0F9           0590     BNE LOOP2
                    0600 ;
                    0610 ;SET UP DISPLAY LIST INTERRUPT VECTOR
```

```
                    0620 ;
06A1 A950           0630      LDA #$50
06A3 8D0002         0640      STA VDSLST
06A6 A906           0650      LDA #$06
06A8 8D0102         0660      STA VDSLST+1
06AB A900           0670      LDA #$00
06AD 85CE           0680      STA COLCNT   INITIALIZE COLOR COUNTER
06AF 85CF           0690      STA DECK     INITIALIZE DECK COUNTER
06B1 8D0FD4         0700      STA NMIRES   RESET INTRPT. STATUS REGISTER
06B4 AD0FD4         0710 WAIT LDA NMIST    GET INTERRUPT STATUS REGISTER
06B7 2940           0720      AND #$40     HAS VERTICAL BLANK OCCURRED?
06B9 F0F9           0730      BEQ WAIT     NO, KEEP CHECKING
06BB AD0ED4         0740      LDA NMIEN    YES, ENABLE DISPLAY LIST.
06BE 0980           0750      ORA #$80
06C0 8D0ED4         0760      STA NMIEN    THIS ENABLES DLI
06C3 60             0770      RTS          ALL DONE
                    0780 ;
                    0790 ;DISPLAY LIST INTERRUPT SERVICE ROUTINE
                    0800 ;
06C4                0810      *=  $0650
0650 48             0820 DLISRV PHA        SAVE ACCUMULATOR
0651 A5CE           0830      LDA COLCNT   GET CURRENT COLOR
0653 18             0840      CLC
0654 6910           0850      ADC #$10     NEXT COLOR
0656 85CE           0860      STA COLCNT   SAVE IT.
0658 D013           0870      BNE OVER     END OF DECK??
                    0880 ;
                    0890 ;END OF DECK, BLACKEN SCREEN
                    0900 ;
065A 8D0AD4         0910      STA WSYNC    WAIT FOR NEXT SCAN LINE
065D 8D1AD0         0920      STA COLBAK   BLACKEN ALL REGISTERS
0660 8D16D0         0930      STA COLPF0
0663 8D17D0         0940      STA COLPF1
0666 8D18D0         0950      STA COLPF2
0669 E6CF           0960      INC DECK     NEXT DECK.
066B 68             0970      PLA          RESTORE ACCUMULATOR
066C 40             0980      RTI          DONE...
                    0990 ;
                    1000 ;PUT OUT NEXT COLOR, WITH FOUR LUMINOSITIES
                    1010 ;
066D A5CF           1020 OVER LDA DECK     UPPER OR LOWER DECK?
066F 2901           1030      AND #$01     MASK OFF RELEVANT BIT
0671 0A             1040      ASL A        SHIFT INTO HIGH LUMINOSITY
0672 0A             1050      ASL A
0673 0A             1060      ASL A
0674 05CE           1070      ORA COLCNT   MERGE WITH COLOR NYBBLE
0676 8D0AD4         1080      STA WSYNC    WAIT FOR NEXT SCAN LINE
0679 8D1AD0         1090      STA COLBAK   STORE COLOR
067C 6902           1100      ADC #$02     NEXT HIGHER LUMINOSITY
067E 8D16D0         1110      STA COLPF0   STORE COLOR
0681 6902           1120      ADC #$02     NEXT HIGHER LUMINOSITY
0683 8D17D0         1130      STA COLPF1   STORE COLOR
0686 6902           1140      ADC #$02     NEXT HIGHER LUMINOSITY
0688 8D18D0         1150      STA COLPF2   STORE COLOR
068B 68             1160      PLA          RESTORE ACCUMULATOR
068C 40             1170      RTI          DONE
```

SYMBOLS

```
=D01A COLBAK          =00CE COLCNT          =D016 COLPF0          =D017 COLPF1          =D018 COLPF2
=00CF DECK             0650 DLISRV          =0230 DSLSTL           068F LOOP1            069A LOOP2
=D40E NMIEN           =D40F NMIRES          =D40F NMIST            066D OVER            =00CC POINTA
 0680 SETUP           =0200 VDSLST           06B4 WAIT             =D40A WSYNC
```

```
028000  .PAGE "GRAPHICS ROUTINES COMPUTE! #21"
028005 ;FROM COMPUTE #21: GRAPHICS ROUTINES
028010 ;
028015 ; ASM,#P,#D:GRAPHICS.OBJ
028020 ; LIST#D:GRAPHICS.SRC
028025 ; CIO EQUATES
028030 ;
028035 CIO = $E456 ; CALL OS THRU HERE
028040 ICCOM = $342 ; COMMAND TO CIO IN IOCB
028045 ICBADR = $344 ; BUFFER OR FILENAME ADDRESS
028050 ICBLEN = $348 ; BUFFER LENGTH
028055 ICAUX1 = $34A ; AUXILLIARY BYTE #1
028060 ICAUX2 = $34B ; AUXIALLARY BYTE #2
028065 COPN = 3 ; COMMAND OPEN
028070 CCLOSE = 12 ; COMMAND CLOSE
028075 CGBINR = 7 ; COMMAND GET BINARY RECORD
028080 CPBINR = 11 ; COMMAND PUT BINARY RECORD
028085 CDRAW = 17 ; COMMAND DRAWTO
028090 CFILL = 18 ; COMMAND FILL (NOT USED IN THIS DEMO.)
028095 OPIN = 4 ; OPEN FOR INPUT
028100 FILDAT = 765 ; WHERE XIO WANTS THE FILL COLOR
028105 CFILL = 18 ; FILL IS XIO 18
028110 OPOUT = 8 ; OPEN FOR OUTPUT
028115 ;
028120 ; EQUATES USED BY THE "S:" DRIVER AND
028125 ;         THE VBLANK ROUTINES
028130 ;
028135 HORIZONTAL = $55
028140 VERTICAL = $54
028145 DRAWCOLOR = $2FB
028150 COLOR0 = $2C4
028155 ;
028160 LOW = $FF
028165 HIGH = $100
028170 RANDOM = $D20A
028175 ;
028180 ;THE ACTUAL ROUTINES
028185 ;
028190 ;
028195  .PAGE "GRAPHICS ROUTINES"
028200 ;
028205 ;FIRST, SET THE LOCATION AND SOME MISC. RAM USAGE
028210 ;
028215 SAVECOLOR .BYTE 0 ;WHERE COLOR IS SAVED
028220 SNAME .BYTE "S:",0  ;THE FILENAME FOR OPEN
028225 ;
028230 ;
028235 ;GRAPHICS G
028240 ;
028245 ;         ENTRY: A-REG CONTAINS GRAPHICS MODE "G"
028250 ;         EXIT:  Y-REG HAS COMPLETION STATUS
028255 ;
028260 GRAPHICS
028265  PHA ; SAVE "G"
028270  LDX #6*$10 ;FILE 6
028275  LDA #CCLOSE
028280  STA ICCOM,X
028285  JSR CIO ;FIRST, WE MUST CLOSE FILE #6
028290 ;(WE IGNORE ANY ERRORS FROM THE CLOSE)
028295 ;
028300  LDX #6*$10 ; AGAIN, FILE 6
028305  LDA #COPN ; WE WILL OPEN THIS "FILE"
028310  STA ICCOM,X
028315  LDA #SNAME&LOW
028320  STA ICBADR,X ; WE USE THE FILE NAM "S:"
```

```
028325  LDA #SNAME/HIGH
028330  STA ICBADR+1,X ; BY POINTING TO IT.
028335  ;
028340  ;ALL IS SET UP FOR OPEN, NOW
028345  ;WE TELL CIO (AND S:) WHAT KIND OF OPEN
028350  ;
028355  PLA ; OUR SAVED "G" GRAPHICS MODE
028360  STA ICAUX2,X ; IS GIVEN TO "S:"
028365  ;
028370  ; (NOTE THAT S: IGNORES THE UPPER BITS OF AUX2)
028375  ;
028380  AND #$F0 ; NOW WE GET JUST THE UPPER BITS
028385  EOR #$10 ; AND FLIP BIT 4
028390  ;
028395  ;(READ THE TEXT. S: EXPECTS THIS BIT INVERTED
028400  ;         FROM WHAT NORMAL BASIC USAGE IS.)
028405  ;
028410  ORA #$0C ; ALLOW READ AND WRITE ACCESS (FOR CIO)
028415  STA ICAUX1,X ; MAKE CIO AND S: HAPPY
028420  JSR CIO ;    AND DO THE OPEN OF S:
028425  RTS
028430  ;
028435  ;         COLOR C
028440  ;
028445  ;         ENTER: COLOR "C" IN A-REGISTER
028450  ;          EXIT: UNCHANGED
028455  ;
028460 COLOR
028465  STA SAVECOLOR
028470  RTS ; EXCITING, WASN'T IT?
028475  ;
028480  ;
028485  ;    POSITION H,V
028490  ;
028495  ;         ENTER: H (HORIZ) POSITION IN X,A REGISTERS (LSB,MSB)
028500  ;                V (VERT) POSITION IN Y-REGISTER
028505  ;
028510  ;          EXIT: UNCHANGED
028515  ;
028520 POSITION
028525  STX HORIZONTAL
028530  STA HORIZONTAL+1 ; READ THE TEXT (MAGAZINE)
028535  STY VERTICAL ; TOO SIMPLE, RIGHT?
028540  RTS
028545  ;
028550  ; PLOT H,V
028555  ;
028560  ;         ENTER: MUST HAVE DONE A PREVIOUS COLOR CALL
028565  ;                X,A, AND Y REGISTERS SET AS IN POSITION
028570  ;
028575  ;
028580 PLOT
028585  JSR POSITION
028590  LDX #6*$10 ; FILE 6, AGAIN
028595  LDA #CPBINR ; COMMAND PUT BINARY RECORD
028600  STA ICCOM,X
028605  LDA #0
028610  STA ICBLEN,X
028615  STA ICBLEN+1,X ; IF BUFFER LENGTH IS ZERO...
028620  LDA SAVECOLOR ; THEN CPBINR PUTS ONE CHAR FROM A-REG
028625  JSR CIO ; AND THIS IS HOW WE PLOT
028630  RTS
028635  ;
028640  ;
028645  ; LOCATE H,V,C
```

```
028655 ;         ENTER: X,A, AND Y REGISTERS SET UP AS IN POSITION
028660 ;          EXIT: A-REG HAS THE LOCATED COLOR
028665 ;                Y-REG HAS THE COMPLETION CODE
028670 ;
028675 LOCATE
028680  JSR POSITION
028685  LDX #6*$10
028690  LDA #CGBINR ; COMMAND GET BINARY RECORD
028695  STA ICCOM,X
028700  LDA #0
028705  STA ICBLEN,X
028710  STA ICBLEN+1,X ; IF BUFFER LENGTH IS ZERO,
028715  JSR CIO ; THEN THE CHAR IS RETURNED IN A
028720  RTS
028725 ;
028730 ;
028735 ;         DRAWTO H,V
028740 ;
028745 ;         ENTER: MUST HAVE DONE A PREVIOUS PLOT
028750 ;                 X,A, AND Y REGISTERS AS IN POSITION
028755 ;
028760 ;          EXIT: Y-REGISTER HAS COMPLETION CODE
028765 ;
028770 ;
028775 DRAWTO
028780  JSR POSITION
028785  LDA SAVECOLOR
028790  STA DRAWCOLOR ; WHERE DRAWTO EXPECTS ITS COLOR
028795  LDX #6*$10 ; FILE #6
028800  LDA #CDRAW ; JUST A COMMAND TO "S:"
028805  STA ICCOM,X
028810  LDA #$0C
028815  STA ICAUX1,X ; INSURANCE
028820  LDA #0
028825  STA ICAUX2,X ; ...GUARANTEED TO WORK
028830  JSR CIO ; DO THE ACTUAL DRAWTO
028835  RTS
028840 ;
028845 ;
028850 ; SETCOLOR REG,HUE,LUM
028855 ;
028860 ;         ENTER: X-REG HAS COLOR REGISTER
028865 ;                A-REG HAS HUE
028870 ;                Y-REG HAS LUMINANCE
028875 ;
028880 ;          EXIT: (UNDEFINED)
028885 ;
028890 SETCOLOR
028895  ASL A
028900  ASL A
028905  ASL A
028910  ASL A ; WE NEED HUE * 16
028915  STA COLOR0,X ; SAVE IT HERE FOR A WHILE
028920  TYA
028925  AND #$0E ; ONLY LUMINANCE BITS THAT MATTER
028930  CLC
028935  ADC COLOR0,X ; END OF THE WHILE
028940  STA COLOR0,X ; AND VBLANK WILL MOVE THIS TO HARDWARE
028945  RTS
028950 ;
028955 ;
028960 ; RND R
028965 ;
028970 ;         ENTER: R IN ACCUMULATOR
028975 ;          EXIT: A-REGISTER CONTAINS A RANDOM
028980 ;                NUMBER FROM 0 TO R, INCLUSIVE
```

```
028985 ;
028990 RND
028995  STA RTEMP
029000  INC RTEMP ; MAKES CMP EASIER
029005 RNDWAIT
029010  LDA RANDOM ; SET A RANDOM NUMBER
029015  CMP RTEMP ; TOO BIG?
029020  BCS RNDWAIT ; YES...WAIT
029025  RTS
029030 ;
029035 RTEMP .BYTE 0 ; RESERVE SPACE
029040 ;
029045 ;
029050 ; FILL ROUTINE
029055 ; ENTER: MUST HAVE PREVIOUSLY DRAWN THE RIGHT HAND EDGE
029060 ; OF THE AREA TO BE FILLED VIA JSR'S TO PLOT AND DRAWTO.
029065 ; JUST PRIOR TO JSR FILL, IT MUST HAVE PERFORMED A JSR PLOT
029070 ; TO ESTABLISH THE TOP (OR BOTTOM OF THE LINE WHICH WILL
029075 ; DEFINE THE LEFT EDGE OF THE AREA TO BE FILLed. FILL PRESUMES THAT THE COLOR TO FILL WITH IS
029080 ; THAT WHICH WAS MOST RECENTLY CHOSEN VIA JSR COLOR. FINALLY,
029085 ; ON ENTRY, FILL EXPECTS THE REGISTERS TO SPECIFY THE
029090 30160 ; ENDING POSITION OF THE LINE WHICH WILL DEFINE THE LEFT
029095 ; EDGE OF THE FILLED AREA, AS FOLLOWS:
029100 ;       HORIZ POS. IN X,A REGISTERS
029105 ;      (X HAS LSB, A HAS MSB)
029110 ;       VERT POS.  IN Y REGISTER
029115 ;
029120 ; EXIT: Y-REGISTER HAS COMPLETION STATUS FROM OS FILL ROUTINE
029125 ;
029130 FILL
029135  JSR POSITION
029140  LDA SAVECOLOR ; ESTABLISHED VIA JSR COLOR
029145  STA FILDAT ; SEE BASIC MANUAL. COLOR USED FOR FILL.
029150  LDX #6*$10 ; FILE 6...WHERE "S:" NORMALLY IS
029155  LDA #CFILL ; XIO 18
029160  STA ICCOM,X ; ... IS SPECIFIED
029165  LDA #0
029170  STA ICAUX1,X ; REMEMBER: XIO,18,#6,0
029175  JSR CIO ; AND, LET THE OS DO ALL THE WORK.
029180  RTS ; ... AND GIVE US STATUS IN THE Y-REG.
```

```
10 ;ATARI 800 SCREEN DUMP UTILITY FOR DUMPING GRAPHICS 8 PICTURES
20 ;TO EPSON MX-80 PRINTERS WITH GRAFTRAX
30 ;BY DAVID NEWCORN 2/28/82
40 ;ASSEMBLY LANG. LISTING (COMPUTE!'S SECOND BOOK OF ATARI)
0100 ADR = $FFFF ;DUMMY ADDRESS (SCREEN MEM ADDR)
0110 STR = $FFFF ;DUMMY ADDRESS (STRING ADDR)
0120  *= $600
0130  PLA ;PULL OFF AUX BYTE FROM BASIC
0140  PLA ;PULL HI BYTE OF STRING STORAGE
0150  STA LOA+2 ;STORE HI BYTE
0160  PLA ;PULL LO BYTE OF STRING STORAGE
0170  STA LOA+1 ;STORE LO BYTE
0180  PLA ; PULL HI BYTE OF BEGINNING OF SCREEN MEMORY
0190  STA STO+2 ; STORE HI BYTE
0200  PLA ; PULL LO BYTE OF SCREEN MEM
0210  STA STO+1 ; STORE LO BYTE
0220  LDY #193 ;LOAD Y AXIS COUNTER
0230 LOA
0240  LDA ADR ; LOAD SCREEN BYTE
0250  DEY ;DECREMENT COUNTER
0260  BEQ RET ;IF DONE THEN RETURN TO BASIC
0270 STO
0280  STA STR ;STORE SCREEN BYTE IN A$
0290  INC STO+1 ;INCREMENT LOW END OF STRING
0300  BEQ BIG1 ;IF LOW END OVERFLOWS, THEN INC HI END
0310 CONT
0320  LDA LOA+1 ;LOAD ACCUM WITH LOW SCREEN ADDRESS
0330  SEC  ; SET UP SUBTRACT W/O CARRY
0340  SBC #40 ;SUBTRACT 40 (40 BYTES PER SCAN LINE)
0350  STA LOA+1 ;STORE RESULT
0360  BCC BIG ; IF UNDERFLOW, DEC HI BYTE OF SCREEN MEM
0370  CLC
0380  JMP LOA ;LOAD NEXT BYTE
0390 BIG
0400  DEC LOA+2 ;DECREMENT HI BYTE OF SCREEN MEM
0410  JMP LOA ;LOAD NEXT BYTE
0420 BIG1
0430  INC STO+2 ;INCREMENT HI BYTE OF STRING STORAGE
0440  JMP CONT ;CONTINUE
0450 RET
0460  RTS ;RETURN TO BASIC
```

```
                    10 ;ATARI 800 SCREEN DUMP UTILITY FOR DUMPING GRAPHICS 8 PICTURES
                    20 ;TO EPSON MX-80 PRINTERS WITH GRAFTRAX
                    30 ;BY DAVID NEWCORN 2/28/82
                    40 ;ASSEMBLY LANG. LISTING (COMPUTE!'S SECOND BOOK OF ATARI)
                    50 ; ASM #D:SCRDMP.SRC,#P:,#D:SCRDMP.OBJ
    =FFFF           60 ADR   =   $FFFF    ;DUMMY ADDRESS (SCREEN MEM ADDR)
    =FFFF           70 STR   =   $FFFF    ;DUMMY ADDRESS (STRING ADDR)
0000                80       *=  $0642
                    90 ; X=USR(1602,.....)
0642 68             0100     PLA          ;PULL OFF AUX BYTE FROM BASIC
0643 68             0110     PLA          ;PULL HI BYTE OF STRING STORAGE
0644 8D5706         0120     STA LOA+2    ;STORE HI BYTE
0647 68             0130     PLA          ;PULL LO BYTE OF STRING STORAGE
0648 8D5606         0140     STA LOA+1    ;STORE LO BYTE
064B 68             0150     PLA          ; PULL HI BYTE OF BEGINNING OF SCREEN MEMORY
064C 8D5D06         0160     STA STO+2    ; STORE HI BYTE
064F 68             0170     PLA          ; PULL LO BYTE OF SCREEN MEM
0650 8D5C06         0180     STA STO+1    ; STORE LO BYTE
0653 A0C1           0190     LDY #193     ;LOAD Y AXIS COUNTER
0655                0200 LOA
0655 ADFFFF         0210     LDA ADR      ; LOAD SCREEN BYTE
0658 88             0220     DEY          ;DECREMENT COUNTER
0659 F023           0230     BEQ RET      ;IF DONE THEN RETURN TO BASIC
065B                0240 STO
065B 8DFFFF         0250     STA STR      ;STORE SCREEN BYTE IN A$
065E EE5C06         0260     INC STO+1    ;INCREMENT LOW END OF STRING
0661 F018           0270     BEQ BIG1     ;IF LOW END OVERFLOWS, THEN INC HI END
0663                0280 CONT
0663 AD5606         0290     LDA LOA+1    ;LOAD ACCUM WITH LOW SCREEN ADDRESS
0666 38             0300     SEC          ; SET UP SUBTRACT W/O CARRY
0667 E928           0310     SBC #40      ;SUBTRACT 40 (40 BYTES PER SCAN LINE)
0669 8D5606         0320     STA LOA+1    ;STORE RESULT
066C 9004           0330     BCC BIG      ; IF UNDERFLOW, DEC HI BYTE OF SCREEN MEM
066E 18             0340     CLC
066F 4C5006         0350     JMP LOA      ;LOAD NEXT BYTE
0672                0360 BIG
0672 CE5706         0370     DEC LOA+2    ;DECREMENT HI BYTE OF SCREEN MEM
0675 4C5006         0380     JMP LOA      ;LOAD NEXT BYTE
0678                0390 BIG1
0678 EE5D06         0400     INC STO+2    ;INCREMENT HI BYTE OF STRING STORAGE
067B 4C6306         0410     JMP CONT     ;CONTINUE
067E                0420 RET
067E 60             0430     RTS          ;RETURN TO BASIC
```

Data 104, 104, 141, 87, 6, 104
" 141, 86, 6, 104, 141, 93, 6
" 104, 141, 92, 6, 160, 193
" 173, 255, 255, 136, 240, 35
" 141, 255, 255, 238, 92, 6
240, 21, 173, 86, 6, 56, 233, 40
141, 86, 6, 144, 4, 24, 76, 85, 6
206, 87, 6, 76, 85, 6, 238,
93, 6, 76, 99, 6, 96

```
0000          0100          .TITLE "MICRO-DOS"
0000          0110                .TAB    20,30,40
              0140 ;     COMPUTE VOL.4,NO.7
              0150 ;        JULY, 1982
              0160 ;      DEFINE EQUATES
              0170 ;   LIST#D:MICRODOS.SRC
0000          0180                *=      $0342
0342          0190 ICCMD          *=      *+2
0352          0200 ICCMD1         =       ICCMD+16
0344          0210 ICBAL          *=      *+1
0354          0220 ICBAL1         =       ICBAL+16
0345          0230 ICBAH          *=      *+3
0355          0240 ICBAH1         =       ICBAH+16
0348          0250 ICBLL          *=      *+1
0358          0260 ICBLL1         =       ICBLL+16
0349          0270 ICBLH          *=      *+1
0359          0280 ICBLH1         =       ICBLH+16
034A          0290 ICAX1          *=      *+1
035A          0300 ICAX11         =       ICAX1+16
034B          0310 ICAX2          *=      *+1
E456          0320 CIOV           =       $E456
0003          0330 OPEN           =       3
000C          0340 CLOSE          =       12
0007          0350 GETC           =       7
000B          0360 PUTC           =       11
0005          0370 GETR           =       5
0009          0380 PUTR           =       9
009B          0390 EOL            =       155
007D          0400 CLR            =       125
007F          0410 TAB            =       127
              0420 ;  MICRO-DOS SOURCE CODE
034C          0430                *=      $0600
              0440 ;
              0450 ;     DISPLAY MENU ON SCREEN
              0460 ;
              0470 INIT
0600 A025     0480                LDY     #MBUF&255
0602 A232     0490                LDX     #EOM-MBUF
0604 A90B     0500                LDA     #PUTC
0606 20CF06   0510                JSR     IOE2
              0520 ;
              0530 ;        ACCEPT COMMAND
              0540 ;
              0550 GRAB
0609 201106   0560                JSR     INPUT
              0570 ;
              0580 ;        EXECUTE COMMAND
              0590 ;
060C 20DB06   0600                JSR     IOP
060F 10EF     0610                BPL     INIT
              0620 MBI GRAB  If CIOV error, Y>127
```

```
)***ERROR - 6
                0630 ;
                0640 ;          INPUT CONTROL ROUTINE
                0650 ;
                0660 INPUT
                0670 ;          PUT PROMPT ON SCREEN
0611 A0F3       0680                LDY     #QBUF&255
0613 A202       0690                LDX     #2
0615 205206     0700                JSR     Q
                0710 ;          GET COMMAND CHOICE
0618 205806     0720                JSR     IN
061B A006       0730                LDY     #6
061D 8C5A03     0740                STY     ICAX1       To "open directory" later
                0750 LOOP
0620 88         0760                DEY
0621 F03F       0770                BEQ     EXIT
0623 B9E806     0780                LDA     ASCII-1,Y
0626 CD0801     0790                CMP     KBUF
0629 D0F5       0800                BNE     LOOP
062B B9ED06     0810                LDA     CMD-1,Y
062E 48         0820                PHA                 Save Command Number
062F C9FE       0830                CMP     #$FE
0631 D013       0840                BNE     CONT
                0850 ;          VERIFY FORMAT COMMAND
0633 A0F8       0860                LDY     #QBUF+5&255
0635 A203       0870                LDX     #3
0637 205206     0880                JSR     Q
063A 205806     0890                JSR     IN
063D AD0801     0900                LDA     KBUF
0640 C944       0910                CMP     #68         Is first letter a "D" ?
0642 D01E       0920                BNE     EXIT        No, then leave MICRO-DOS
0644 F00A       0930                BEQ     RT
                0940 CONT
                0950 ;          PUT PROMPT ON SCREEN
0646 A0F5       0960                LDY     #QBUF+2&255
0648 A203       0970                LDX     #3
064A 205206     0980                JSR     Q
                0990 ;          GET FILENAME
064D 205806     1000                JSR     IN
                1010 RT
0650 68         1020                PLA                 Retrieve FMS command
0651 60         1030                RTS
                1040 ;
                1050 ;          PROMPT ON SCREEN ROUTINE
                1060 ;
                1070 Q
0652 A90B       1080                LDA     #PUTC
0654 20B806     1090                JSR     IOE
0657 60         1100                RTS
                1110 ;
                1120 ;          EDITOR LINE IN ROUTINE
                1130 ;
```

```
                1140 IN
0658 A008       1150            LDY     #KBUF&255
065A A228       1160            LDX     #40
065C A905       1170            LDA     #GETR
065E 20CF06     1180            JSR     IOE2
0661 60         1190            RTS
                1200 ;
                1210 ;      EXIT AND I/O ROUTINES
                1220 ;
                1230 EXIT
0662 AD0801     1240            LDA     KBUF
0665 C94D       1250            CMP     #77     Directory Request ??
0667 D03E       1260            BNE     DOS     No, so to DOS
0669 A0F9       1270            LDY     #QBUF+8&255
066B A203       1280            LDX     #3
066D 205206     1290            JSR     Q       Put D#? prompt on screen
0670 A001       1300            LDY     #SBUF+1&255
0672 A201       1310            LDX     #1
0674 A905       1320            LDA     #GETR
0676 20CF06     1330            JSR     IOE2    Get drive number
0679 A03A       1340            LDY     #58
067B 8C0201     1350            STY     SBUF+2  Restore colon
067E A000       1360            LDY     #SBUF&255
0680 A903       1370            LDA     #OPEN
0682 20DD06     1380            JSR     IOP2    Open directory
                1390 AGAIN
0685 A913       1400            LDA     #19
0687 8D5803     1410            STA     ICBLL1
068A A905       1420            LDA     #GETR
068C 20DB06     1430            JSR     IOP     Get directory entry
068F 300B       1440            BMI     THRU    Last entry? Go to THRU
0691 A007       1450            LDY     #KBUF-1&255     Tab is first characte
0693 A214       1460            LDX     #20
0695 A909       1470            LDA     #PUTR
0697 20CF06     1480            JSR     IOE2    Put directory entry on screen
069A 10E9       1490            BPL     AGAIN
                1500 THRU
069C A90C       1510            LDA     #CLOSE
069E 20DB06     1520            JSR     IOP     Close directory
06A1 205806     1530            JSR     IN
06A4 4C0006     1540            JMP     INIT    Go to Selection Menu
                1550 ;
                1560 DOS
06A7 C941       1570            CMP     #65
06A9 D00A       1580            BNE     BASIC
06AB A944       1590            LDA     #68
06AD CD0901     1600            CMP     KBUF+1  Additional check for safety
06B0 D003       1610            BNE     BASIC
06B2 4C9F17     1620            JMP     6047    Load DUP.SYS
                1630 ;
                1640 BASIC
06B5 4C4DA0     1650            JMP     $A04D   Warm Start entry
```

```
06F9 44
06FA 23
06FB 3F
                2050 ;      PUT THE REST IN PAGE 1
06FC            2060            *=      $100
                2070 SBUF
0100 44         2080            .BYTE   "D1:*.*",EOL,TAB
0101 31
0102 3A
0103 2A
0104 2E
0105 2A
0106 9B
0107 7F
                2090 KBUF
0108            2100            *=      *+29
                2110 MBUF
0125 7D         2120            .BYTE   CLR,204,"OCK",EOL
0126 CC
0127 4F
0128 43
0129 4B
012A 9B
012B D5         2130            .BYTE   213,"NLOCK",EOL
012C 4E
012D 4C
012E 4F
012F 43
0130 4B
0131 9B
0132 C4         2140            .BYTE   196,"ELETE",EOL
0133 45
0134 4C
0135 45
0136 54
0137 45
0138 9B
0139 D2         2150            .BYTE   210,"ENAME",EOL
013A 45
013B 4E
013C 41
013D 4D
013E 45
013F 9B
0140 C6         2160            .BYTE   198,"ORMAT",EOL
0141 4F
0142 52
0143 4D
0144 41
0145 54
0146 9B
0147 CD         2170            .BYTE   205,"ENU",EOL
```

```
                1660 ;
                1670 IOE
06B8 48         1680            PHA
06B9 A906       1690            LDA     #QBUF/256
06BB 8D4503     1700            STA     ICBAH
06BE 68         1710            PLA
                1720 ENT
06BF 8C4403     1730            STY     ICBAL
06C2 8E4803     1740            STX     ICBLL
06C5 A200       1750            LDX     #0
06C7 8E4903     1760            STX     ICBLH
06CA 8E5903     1770            STX     ICBLH1
06CD F013       1780            BEQ     IO
                1790 IOE2
06CF 48         1800            PHA
06D0 A901       1810            LDA     #KBUF/256
06D2 8D4503     1820            STA     ICBAH
06D5 8D5503     1830            STA     ICBAH1
06D8 68         1840            PLA
06D9 D0E4       1850            BNE     ENT
                1860 ;
                1870 IOP
06DB A008       1880            LDY     #KBUF&255
                1890 IOP2
06DD 8C5403     1900            STY     ICBAL1
06E0 A210       1910            LDX     #$10
                1920 IO
06E2 9D4203     1930            STA     ICCMD,X
06E5 2056E4     1940            JSR     CIOV
06E8 60         1950            RTS
                1960 ;
                1970 ;      DEFINE MENU AND BUFFERS
                1980 ASCII
06E9 4C         1990            .BYTE   76,85,68,82,70
06EA 55
06EB 44
06EC 52
06ED 46
                2000 CMD
06EE 23         2010            .BYTE   $23,$24,$21,$20,$FE
06EF 24
06F0 21
06F1 20
06F2 FE
                2020 ;
                2030 QBUF
06F3 1D         2040            .BYTE   29,">FN?",198,"D#?"
06F4 3E
06F5 46
06F6 4E
06F7 3F
06F8 C6
```

MICRO-DOS

```
0148 45
0149 4E
014A 55
014B 9B
014C C1      2180            .BYTE   193,196,"OS",EOL
014D C4
014E 4F
014F 53
0150 9B
0151 C2      2190            .BYTE   194,"ASIC",EOL
0152 41
0153 53
0154 49
0155 43
0156 9B
             2200 EOM
0157         2210            *=      10
000A 00      2220            .BYTE   0,6
000B 06
000C         2230            *=      5446
1546 00      2240            .BYTE   0
1547         2250            *=      5450
154A 06      2260            .BYTE   6
154B         2270            .END
```

```
                10 ;ROUTINE EXCLOR
                20 ;PERFORMS EXCLUSIVE OR OPERATION ON
0000            30          .TITLE          EXCLOR
0000            40              .TAB 15,20,35
                50 ;TWO BYTES PASSED THROUGH THE STACK
                60 ;PASSES RESULTS DIRECTLY THROUGH USR FUNCTION
                70 ;
0000            80              *=    $0600
00CC            90 TEMPL        =     $CC             TEMP HOLDING LOCATION
00CD            0100 TEMPH      =     $CD             TEMP HOLDING LOCATION
00D4            0110 RESLTL     =     $D4             ADDRESS FOR PASSING RESULTS
00D5            0120 RESLTH     =     $D5             ADDRESS FOR PASSING HIGH RESULT
0600 68         0130 EXCLOR     PLA
0601 68         0140            PLA
0602 85CD       0150            STA   TEMPH           SAVE HIGH BYTE
0604 68         0160            PLA
0605 85CC       0170            STA   TEMPL           SAVE LOW BYTE
0607 68         0180            PLA
0608 45CD       0190            EOR   TEMPH           PERFORM HIGH EXCLUSIVE OR
060A 85D5       0200            STA   RESLTH          STORE RESULT
060C 68         0210            PLA
060D 45CC       0220            EOR   TEMPL           PERFORM LOW EXCLUSIVE OR
060F 85D4       0230            STA   RESLTL          STORE RESULT
0611 60         0240            RTS                   WHAT COULD BE SIMPLER?
0612            0250            .END
```

=00CC TEMPL =00CD TEMPH =00D4 RESLTL =00D5 RESLTH
0600 EXCLOR

```
0000                    .TITLE          ROUTINE NOTE
                        ;
                        ;FROM ATARI ASSEMBLER REF MANUAL
                        ;PP.68-70 (EX.2)
                        ;
0000                            .TAB 15,20,35
                        ;GENERATES NOTES WITH CONTROLLABLE ATTACK AND DECAY TIMES
                        ;CALL FROM BASIC WITH COMMAND:
                        ;A=USR(1536,F,A,P,D)
                        ;WHERE
                        ;F IS THE FREQUENCY
                        ;A IS THE ATTACK TIME
                        ;P IS THE PEAK TIME
                        ;D IS THE DECAY TIME
                        ;
                        ;ALL TIMES GIVEN IN UNITS OF 1.6 MILLISECONDS
                        ;
0000                    *=      $0600
D200            AUDF1   =       $D200           AUDIO FREQ REGISTER
D201            AUDC1   =       $D201           AUDIO CONTROL REGISTER
00CC            ATTACK  =       $CC             ATTACK TIME
00CD            PEAK    =       $CD             PEAK OR PLATEAU TIME
00CE            DECAY   =       $CE             DECAY TIME
0600 68         NOTE    PLA
0601 68                 PLA
0602 68                 PLA
0603 8D00D2             STA     AUDF1           SET FREQUENCY
0606 68                 PLA
0607 85CC               STA     ATTACK          SET ATTACK TIME
0609 68                 PLA
060A 68                 PLA
060B 68                 PLA
060C 85CD               STA     PEAK            SET PEAK TIME
060E 68                 PLA
060F 68                 PLA
0610 85CE               STA     DECAY           SET DECAY TIME
                        ;
                        ;ATTACK LOOP
                        ;
0612 A9A0               LDA     #$A0            START WITH ZERO VOLUME
0614 8D01D2     ATLOOP  STA     AUDC1
0617 A6CC               LDX     ATTACK
0619 204106             JSR     DELAY
061C 18                 CLC
061D 6901               ADC     #$01
061F C9B0               CMP     #$B0
0621 D0F1               BNE     ATLOOP
                        ;
                        ;PEAK LOOP
                        ;
0623 A90E               LDA     #$0E
```

```
0625 A6CD   0520 PKLOOP    LDX  PEAK
0627 204106 0530           JSR  DELAY
062A 38     0540           SEC
062B E901   0550           SBC  #$01
062D D0F6   0560           BNE  PKLOOP
062F A9AF   0570           LDA  #$AF
0631 8D01D2 0580 DCLOOP    STA  AUDC1
0634 A6CE   0590           LDX  DECAY
0636 204106 0600           JSR  DELAY
0639 38     0610           SEC
063A E901   0620           SBC  #$01
063C C99F   0630           CMP  #$9F
063E D0F1   0640           BNE  DCLOOP
0640 60     0650           RTS
0641 A013   0660 DELAY     LDY  #$13
0643 88     0670 DELAY2    DEY
0644 D0FD   0680           BNE  DELAY2
0646 CA     0690           DEX
0647 D0F8   0700           BNE  DELAY
0649 60     0710           RTS
064A        0720           .END
```

```
=D200 AUDF1          =D201 AUDC1          =00CC ATTACK         =00CD PEAK
=00CE DECAY           0600 NOTE            0614 ATLOOP          0641 DELAY
 0625 PKLOOP          0631 DCLOOP          0643 DELAY2
```

```
0000          10            .TITLE "SPLAY"
0000          20              .TAB 15,20,35
              30 ;FROM ATARI MANUAL PP.70-72.
              40 ;ROUTINE SPLAY
              50 ;PUTS A PRETTY DISPLAY ONTO THE SCREEN
              60 60 ;CALL FROM BASIC WITH THE FOLLOWING COMMANDS
)***ERROR - 6-SYNTAX
              70 ;GR.19:A=USR(1536)
              80 ;EXIT PROGRAM WITH S RESET
              90 ;
              0100 ;
0000          0110          *=    $0600
00CC          0120 TEMP     =     $CC             TEMPORARY LOCATION
00CD          0130 XLOC     =     $CD             HORIZONTAL POSITION OF PIXEL
00CE          0140 YLOC     =     $CE             VERTICAL POSITION OF PIXEL
00CF          0150 DIST     =     $CF             DIST. OF PIXEL FROM SCREEN CENTER
00D0          0160 PHASE    =     $D0             COLOR PHASE
00D1          0170 COLOR    =     $D1             COLOR CHOICE
0058          0180 SAVMSC   =     $58             POINTER TO BEG. OF DISPLAY MEMORY
02C4          0190 COLOR0   =     $02C4           LOCATION OF COLOR REGISTERS
D20A          0200 RANDOM   =     $D20A           HARDWARE RANDOM NUMBER LOCATION
0600 68       0210 SPLAY    PLA                   POP A ZERO FROM STACK
0601 85D0     0220          STA   PHASE           STORE IT IN PHASE
0603 AA       0230          TAX                   SET COUNTER
              0240 ;THIS IS THE MAIN PROGRAM LOOP
              0250 ;FIRST WE RANDOMLY CHOOSE THE SCREEN LOC. TO MODIFY
              0260 ;SCREEN IS 40 PIXELS HORIZONTALLY BY 24 PIXELS VERTICALLY
              0270 ;WITH 4 HORIZONTALLY ADJACENT PIXELS PER BYTE
              0280 ;HENCE THERE ARE 10 BYTES PER HORIZONTAL ROW
              0290 ;AND 24 ROWS FOR A TOTAL OF 240 BYTES
              0300 ;TO REPRESENT THE SCREEN
              0310 ;
              0320 ;
              0330 ;
0604 AD0AD2   0340 BEGIN    LDA   RANDOM          GET A RANDOM NUMBER
0607 290F     0350          AND   #$0F            MASK OFF LOWER NYBBLE
0609 C90A     0360          CMP   #$0A            MUST BE SMALLER THAN 10
060B B0F7     0370          BCS   BEGIN           IF NOT, TRY AGAIN
060D 85CD     0380          STA   XLOC            STORE THE RESULT
060F 38       0390          SEC
0610 E905     0400          SBC   #$05            GET X-DISTANCE FROM CENTER
0612 1005     0410          BPL   XA              IS IT POSITIVE OR NEGATIVE?
0614 49FF     0420          EOR   #$FF            IF NEGATIVE, MAKE IT POSITIVE
0616 18       0430          CLC
0617 6901     0440          ADC   #$01
0619 85CF     0450 XA       STA   DIST            SAVE THE ABSOLUTE VALUE
061B AD0AD2   0460 TRYAGN   LDA   RANDOM
061E 291F     0470          AND   #$1F            MASK OFF LOWER 5 BITS
0620 C918     0480          CMP   #$18            MUST BE SMALLER THAN 24
0622 B0F7     0490          BCS   TRYAGN          BECAUSE THERE ARE ONLY 24 ROWS.
0624 85CE     0500          STA   YLOC            STORE RESULT.
```

```
SPLAY

0626 38      0510            SEC
0627 E90C    0520            SBC  #$0C            GET Y-DIST FROM CENTER OF SCREEN
0629 1005    0530            BPL  XB              IS IT POSITIVE OR NEGATIVE?
062B 49FF    0540            EOR  #$FF            IF NEGATIVE, MAKE IT POSITIVE
062D 18      0550            CLC
062E 6901    0560            ADC  #$01
             0570 ;
             0580 ;NOW CALCULATE THE COLOR TO PUT INTO THIS POSITION
             0590 ;
0630 18      0600 XB         CLC
0631 65CF    0610            ADC  DIST            TOTAL DIST FROM CENTER OF SCREEN
0633 65D0    0620            ADC  PHASE           COLOR PHASE OFFSET
             0630 ;
             0640 ;BITS 3 AND 4 NOW GIVE THE COLOR TO USE.
             0650 ;
0635 291F    0660            AND  #$1F            MASK OUT BITS 5, 6, AND 7
0637 4A      0670            LSR  A
0638 4A      0680            LSR  A
0639 4A      0690            LSR  A               SHIFT OFF BITS 0, 1, AND 2
063A 85D1    0700            STA  COLOR           STORE RIGHT-JUSTIFIED RESULT
             0710 ;
             0720 ;NOW WE MUST DETERMINE WHICH OF THE 4 PIXELS
             0730 ;IN THE BYTE GET THE COLOR
             0740 ;
063C AD0AD2  0750            LDA  RANDOM
063F 2903    0760            AND  #$03            GET RANDOM NO. BETWEEN 0 AND 3,
0641 A8      0770            TAY                  USE IT AS A COUNTER.
0642 F007    0780            BEQ  NOSHFT          SKIP AHEAD IF IT IS ZERO
             0790 ;
             0800 ;SHIFT OVER TWICE FOR EACH STEP IN Y
             0810 ;
0644 06D1    0820 SHFTLP     ASL  COLOR
0646 06D1    0830            ASL  COLOR
0648 88      0840            DEY
0649 D0F9    0850            BNE  SHFTLP
             0860 ;
             0870 ;NOW WE MUST CALCULATE WHERE IN MEMORY TO PUT OUR SQUARE
             0880 ;
064B A5CE    0890 NOSHFT     LDA  YLOC            GET VERTICAL POSITION
064D 0A      0900            ASL  A               YLOC*2
064E 85CC    0910            STA  TEMP            SAVE IT FOR A FEW MICROSECONDS
0650 0A      0920            ASL  A
0651 0A      0930            ASL  A               YLOC*8
0652 65CC    0940            ADC  TEMP            ADD IN YLOC*2
             0950 ;
             0960 ;RESULTS IN ACCUMULATOR IS YLOC*10
             0970 ;REMEMBER, THERE ARE TEN BYTES PER SCREEN ROW
             0980 ;
0654 65CD    0990            ADC  XLOC
             1000 ;
             1010 ;RESULT IS MEMORY LOCATION OF DESIRED PIXEL GROUP
0656 A8      1020            TAY
```

SPLAY

```
0657 A5D1    1030            LDA  COLOR          GET COLOR BYTE
0659 9158    1040            STA  (SAVMSC),Y     PUT IT ONTO THE SCREEN
065B CA      1050            DEX                 WE SHALL PUT 254 MORE SQUARES
065C D0A6    1060            BNE  BEGIN          ONTO THE SCREEN.
             1070 ;
             1080 ;END OF MAIN INNER LOOP
             1090 ;
065E C6D0    1100            DEC  PHASE          SET COLOR PHASE FOR EXPLOSION.
0660 A5D0    1110            LDA  PHASE
0662 291F    1120            AND  #$1F           EVERY 32 PHASE STEPS
0664 D09E    1130            BNE  BEGIN          WE CHANGE COLOR REGISTERS
             1140 ;THIS SECTION USES BITS 5 AND 6 OF PHASE
             1150 ;TO CHOOSE WHICH COLOR REGISTER TO MODIFY
             1160 ;
0666 A5D0    1170            LDA  PHASE
0668 4A      1180            LSR  A
0669 4A      1190            LSR  A
066A 4A      1200            LSR  A
066B 4A      1210            LSR  A
066C 4A      1220            LSR  A
066D 2903    1230            AND  #$03
066F AA      1240            TAX
             1250 ;
0670 AD0AD2  1260            LDA  RANDOM         CHOOSE A RANDOM COLOR
0673 9DC402  1270            STA  COLOR0,X       PUT NEW COLOR INTO COLOR REGISTER
0676 4C0406  1280            JMP  BEGIN          START ALL OVER
0679         1290            .END
```

SPLAY

=00CC TEMP	=00CD XLOC	=00CE YLOC	=00CF DIST
=00D0 PHASE	=00D1 COLOR	=0058 SAVMSC	=02C4 COLORO
=D20A RANDOM	0600 SPLAY	0604 BEGIN	0617 XA
061B TRYAGN	0630 XB	064B NOSHFT	0644 SHFTLP

```
0000                    .TITLE "***** KATHY'S COLOR PALLETTE *****"
                        ;PUTS ALL 128 COLORS ONTO THE SCREEN
                        ;FROM ATARI ASSEMBLER/EDITOR REFERENCE MANUAL
                        ;PP.72-74
                        ;CALL FROM BASIC WITH THE FOLLOWING COMMANDS:
                        ;GRAPHICS 5
                        ;FOR I=0 TO 3:COLOR I:FOR J=20*I TO 20*I+19:PLOT J,3
                        ;DRAWTO J,39:NEXT J:NEXT I
                        ;A=USR(1536)
                        ;
                        ;BASIC IS STILL USABLE
                        ;EXIT WITH SYSTEM RESET KEY
                        ;
0000                    *=    $0600
00CC            POINTA  =     $CC         POINTER TO DISPLAY LIST
00CE            COLCNT  =     $CE         KEEPS TRACK OF COLOR WE ARE ON
00CF            DECK    =     $CF         BIT 0 KEEPS TRACK OF WHICH DECK
0230            DSLSTL  =     $0230       O.S. DISPLAY LIST ADDRESS
D40E            NMIEN   =     $D40E       NON-MASKABLE INTERRUPT ENABLE
D40F            NMIRES  =     $D40F       NON-MASKABLE INTERRUPT RESET
D40F            NMIST   =     $D40F       NON-MASKABLE INTERRUPT STATUS
0200            VDSLST  =     $0200       DISPLAY LIST INTERRUPT VECTOR
D01A            COLBAK  =     $D01A       BACKGROUND COLOR REGISTER
D016            COLPF0  =     $D016       COLOR REGISTER #0
D017            COLPF1  =     $D017       COLOR REGISTER #1
D018            COLPF2  =     $D018       COLOR REGISTER #2
D40A            WSYNC   =     $D40A       WAIT FOR HORIZONTAL SYNC
0600 68         SETUP   PLA               CLEAN STACK
                        ;SET UP POINTER ON PAGE ZERO
                        ;
0601 AD3002             LDA   DSLSTL
0604 85CC               STA   POINTA
0606 AD3102             LDA   DSLSTL+1
0609 85CD               STA   POINTA+1
                        ;
060B A007               LDY   #$07          POINT TO 3RD MODE BYTE
060D A98A               LDA   #$8A          NEW MODE BYTE
                        ;
                        ;STORE 16 DISPLAY LIST INTERRUPT MODE BYTES
                        ;
060F 91CC       LOOP1   STA   (POINTA),Y
0611 C8                 INY
0612 C017               CPY   #$17
0614 D0F9               BNE   LOOP1
                        ;
                        ;SKIP FOUR BLANK LINES
                        ;
0616 C8                 INY
0617 C8                 INY
0618 C8                 INY
0619 C8                 INY
```

```
***** KATHY'S COLOR PALLETTE *****

                0520 ;
                0530 ;STORE 16 MORE DL INTERRUPT MODE BYTES
                0540 ;
061A 91CC       0550 LOOP2  STA  (POINTA),Y
061C C8         0560        INY
061D C02B       0570        CPY  #$2B
061F D0F9       0580        BNE  LOOP2
                0590 ;
                0600 ;SET UP DISPLAY LIST INTERRUPT VECTOR
                0610 ;
0621 A950       0620        LDA  #$50
0623 8D0002     0630        STA  VDSLST
0626 A906       0640        LDA  #$06
0628 8D0102     0650        STA  VDSLST+1
062B A900       0660        LDA  #$00
062D 85CE       0670        STA  COLCNT      INITIALIZE COLOR COUNTER
062F 85CF       0680        STA  DECK        INITIALIZE DECK COUNTER
0631 8D0FD4     0690        STA  NMIRES      RESET INTRPT. STATUS REGISTER
0634 AD0FD4     0700 WAIT   LDA  NMIST     GET INTERRUPT STATUS REGISTER
0637 2940       0710        AND  #$40        HAS VERTICAL BLANK OCCURRED?
0639 F0F9       0720        BEQ  WAIT        NO, KEEP CHECKING
063B AD0ED4     0730        LDA  NMIEN       YES, ENABLE DISPLAY LIST.
063E 0980       0740        ORA  #$80
0640 8D0ED4     0750        STA  NMIEN       THIS ENABLES DLI
0643 60         0760        RTS             ALL DONE
                0770 ;
                0780 ;DISPLAY LIST INTERRUPT SERVICE ROUTINE
                0790 ;
0644            0800        *=   $0650
0650 48         0810 DLISRV PHA             SAVE ACCUMULATOR
0651 A5CE       0820        LDA  COLCNT      GET CURRENT COLOR
0653 18         0830        CLC
0654 6910       0840        ADC  #$10        NEXT COLOR
0656 85CE       0850        STA  COLCNT      SAVE IT.
0658 D013       0860        BNE  OVER        END OF DECK??
                0870 ;
                0880 ;END OF DECK, BLACKEN SCREEN
                0890 ;
065A 8D0AD4     0900        STA  WSYNC       WAIT FOR NEXT SCAN LINE
065D 8D1AD0     0910        STA  COLBAK      BLACKEN ALL REGISTERS
0660 8D16D0     0920        STA  COLPF0
0663 8D17D0     0930        STA  COLPF1
0666 8D18D0     0940        STA  COLPF2
0669 E6CF       0950        INC  DECK        NEXT DECK.
066B 68         0960        PLA             RESTORE ACCUMULATOR
066C 40         0970        RTI             DONE...
                0980 ;
                0990 ;PUT OUT NEXT COLOR, WITH FOUR LUMINOSITIES
                1000 ;
066D A5CF       1010 OVER   LDA  DECK       UPPER OR LOWER DECK?
066F 2901       1020        AND  #$01        MASK OFF RELEVANT BIT
0671 0A         1030        ASL  A           SHIFT INTO HIGH LUMINOSITY
```

***** KATHY'S COLOR PALLETTE *****

```
0672 0A      1040          ASL   A
0673 0A      1050          ASL   A
0674 05CE    1060          ORA   COLCNT        MERGE WITH COLOR NYBBLE
0676 8D0AD4  1070          STA   WSYNC         WAIT FOR NEXT SCAN LINE
0679 8D1AD0  1080          STA   COLBAK        STORE COLOR
067C 6902    1090          ADC   #$02          NEXT HIGHER LUMINOSITY
067E 8D16D0  1100          STA   COLPF0        STORE COLOR
0681 6902    1110          ADC   #$02          NEXT HIGHER LUMINOSITY
0683 8D17D0  1120          STA   COLPF1        STORE COLOR
0686 6902    1130          ADC   #$02          NEXT HIGHER LUMINOSITY
0688 8D18D0  1140          STA   COLPF2        STORE COLOR
068B 68      1150          PLA                 RESTORE ACCUMULATOR
068C 40      1160          RTI                 DONE
```

```
***** KATHY'S COLOR PALLETTE *****

=00CC POINTA     =00CE COLCNT     =00CF DECK       =0230 DSLSTL
=D40E NMIEN      =D40F NMIRES     =D40F NMIST      =0200 VDSLST
=D01A COLBAK     =D016 COLPF0     =D017 COLPF1     =D018 COLPF2
=D40A WSYNC       0600 SETUP       060F LOOP1       061A LOOP2
 0634 WAIT        0650 DLISRV      066D OVER
```

```
10 ;COMPUTE!'S SECOND BOOK OF ATARI
20 ;
30 ;
40 ;PP.232-235 (BY ROBERT E. ALLEGER)
50 ;
60 ; THIS PROGRAM ALLOWS A BASIC
70 ;PROGRAM TO LOAD A MACHINE LANGUAGE
80 ;PROGRAM AND EXECUTE IT.
90 ;
0100 ;
0110 ;
0120 ; ASM ,#P,#D:LOADIT.OBJ
0130 ; LIST#D:LOADIT.SRC
```

```
0000          0140 *       .PAGE "LOADIT  PROGRAM LISTING"
              0150 ;
              0160 ;        PROG EQUATES
              0170 ;
              0180 ;
0010          0190 IOCB1  =    1*16      ; IOCB#1(D:)
0340          0200 ICHID  =    $0340     ; HANDLER ID
0341          0210 ICDNO  =    ICHID+1   ; DEVICE #
0342          0220 ICCOM  =    ICDNO+1   ; COMMAND
0343          0230 ICSTA  =    ICCOM+1   ; STATUS
0344          0240 ICBAL  =    ICSTA+1   ; BUFFER ADDR
0346          0250 ICPTL  =    ICBAL+2   ; PUT ROUTINE ADDR - 1
0348          0260 ICBLL  =    ICPTL+2   ; BUFFER LENGTH
034A          0270 ICAX1  =    ICBLL+2   ; AUX 1
034B          0280 ICAX2  =    ICAX1+1   ; AUX 2
034C          0290 ICAX3  =    ICAX2+1   ; AUX 3
034D          0300 ICAX4  =    ICAX3+1   ; AUX 4
034E          0310 ICAX5  =    ICAX4+1   ; AUX 5
034F          0320 ICAX6  =    ICAX5+1   ; AUX 6
              0330 ;
E456          0340 CIO    =    $E456
0003          0350 ENR    =    $03       ; EOF ON NEXT READ
0088          0360 EOF    =    $88       ; EOF STATUS
0003          0370 OPEN   =    $03       ; OPEN COMMAND
0007          0380 GETCHR =    $07       ; GET CHARACTERS COMMAND
000C          0390 CLOSE  =    $0C       ; CLOSE COMMAND
0004          0400 OREAD  =    $04       ; OPEN DIRECTION = READ
02E0          0410 RUNLOC =    $02E0     ; AUTO-RUN VECTOR
00CB          0420 FREE0  =    $00CB     ; FREE 0 PAGE RAM (TO $00D1)
00CB          0430 HEADER =    FREE0     ; BLOCK HEADER BUFFER
00CF          0440 FLAG   =    HEADER+4  ;1ST BLOCK FLAG
```

```
BEGINNING

0000            0450          .PAGE "BEGINNING"
                0460 ;
0000            0470          *=   $0600
                0480 ;
                0490 ;          INITIALIZATION
                0500 ;
0600 A210       0510 INIT     LDX  #IOCB1
0602 20AD06     0520          JSR  CLOSEIT    ; IN CASE IOCB WAS IN USE
0605 86CF       0530          STX  FLAG
0607 68         0540          PLA
0608 68         0550          PLA             ; MSB OF FILE SPEC LOCATION
0609 9D4503     0560          STA  ICBAL+1,X
060C 68         0570          PLA             ; LSB OF FILE SPEC LOCATION
060D 9D4403     0580          STA  ICBAL,X
0610 A904       0590          LDA  #OREAD
0612 9D4A03     0600          STA  ICAX1,X
0615 A903       0610          LDA  #OPEN
0617 9D4203     0620          STA  ICCOM,X
061A 2056E4     0630          JSR  CIO        ; OPEN FILE
061D 1003       0640          BPL  RDHDR
061F 4CA606     0650          JMP  ERROR      ; FILE NOT FOUND
                0660 ;
```

```
HEADER I.D. OR START ADDRESS

0622          0670          .PAGE "HEADER I.D. OR START ADDRESS"
              0680 ;
0622 A9CB     0690 RDHDR    LDA  #HEADER&255
0624 9D4403   0700          STA  ICBAL,X
0627 A900     0710          LDA  #HEADER/256
0629 9D4503   0720          STA  ICBAL+1,X
062C A902     0730          LDA  #2
062E 9D4803   0740          STA  ICBLL,X
0631 A900     0750          LDA  #0
0633 9D4903   0760          STA  ICBLL+1,X
0636 A907     0770          LDA  #GETCHR
0638 9D4203   0780          STA  ICCOM,X
063B 2056E4   0790          JSR  CIO        ; GET ID OR START ADDR
063E 1006     0800          BPL  CHKID
0640 C088     0810          CPY  #EOF
0642 F05C     0820          BEQ  DONE
0644 D060     0830          BNE  ERROR
              0840 ;
0646 A9FF     0850 CHKID    LDA  #$FF
0648 C5CB     0860          CMP  HEADER
064A D004     0870          BNE  CONT
064C C5CC     0880          CMP  HEADER+1
064E F0D2     0890          BEQ  RDHDR      ; IGNORE $FF,$FF ID CODE
              0900 ;
              0910 ;READ END ADDRESS
              0920 ;
0650 A9CD     0930 CONT     LDA  #HEADER+2&255
0652 9D4403   0940          STA  ICBAL,X
0655 A900     0950          LDA  #HEADER+2/256
0657 9D4503   0960          STA  ICBAL+1,X
065A 2056E4   0970          JSR  CIO        ; GET END ADDRESS
065D 1002     0980          BPL  HDROK
065F 3045     0990          BMI  ERROR      ; BAD FILE
              1000 ;
              1010 ;STORE PROGRAM START ADDRESS
              1020 ;
0661 A5CF     1030 HDROK    LDA  FLAG
0663 F00E     1040          BEQ  RDBLOK     ; SKIP IF NOT 1ST BLOCK
0665 A5CB     1050          LDA  HEADER     ; USE START OF PROGRAM
0667 8DE002   1060          STA  RUNLOC     ; AS DEFAULT RUN ADDRESS
066A A5CC     1070          LDA  HEADER+1
066C 8DE102   1080          STA  RUNLOC+1
066F A900     1090          LDA  #0
0671 85CF     1100          STA  FLAG       ; CLEAR 1ST BLOCK FLAG
              1110 ;
              1120 ;
              1130 ; READ A BLOCK OF OBJECT CODE
              1140 ;
0673 A5CB     1150 RDBLOK   LDA  HEADER     ; LOAD ADDRESS
0675 9D4403   1160          STA  ICBAL,X
0678 A5CC     1170          LDA  HEADER+1
```

```
HEADER I.D. OR START ADDRESS

067A 9D4503 1180          STA  ICBAL+1,X
067D A5CD   1190          LDA  HEADER+2   ; END ADDRESS
067F 38     1200          SEC
0680 E5CB   1210          SBC  HEADER     ; LENGTH = END-START
0682 9D4803 1220          STA  ICBLL,X
0685 A5CE   1230          LDA  HEADER+3
0687 E5CC   1240          SBC  HEADER+1
0689 9D4903 1250          STA  ICBLL+1,X
068C FE4803 1260          INC  ICBLL,X    ; ADJUST LENGTH BY 1
068F D003   1270          BNE  *+5
0691 FE4903 1280          INC  ICBLL+1,X
            1290 ;
0694 2056E4 1300          JSR  CIO        ; READ BLOCK
0697 1089   1310          BPL  RDHDR      ; GET NEXT BLOCK
0699 C003   1320          CPY  #ENR
069B F085   1330          BEQ  RDHDR      ; THIS IS ALSO OK
069D 4CA606 1340          JMP  ERROR      ; BAD FILE
            1350 ;
```

SUBROUTINES

```
06A0            1360         .PAGE "SUBROUTINES"
06A0            1370         .TITLE "LOADIT.OBJ"
                1380 ;
                1390 ;*****************************************************
                1400 ;
                1410 ;
                1420 ;START SELECTED PROGRAM
                1430 ;
                1440 ;*****************************************************
06A0 20AD06     1450 DONE    JSR  CLOSEIT
06A3 6CE002     1460         JMP  (RUNLOC)  ; START PROGRAM
                1470 ;
                1480 ;*****************************************************
                1490 ;
                1500 ;
                1510 ;RETURN ERROR CODE TO BASIC
                1520 ;
                1530 ;
                1540 ;*****************************************************
                1550 ;
06A6 98         1560 ERROR   TYA
06A7 85D4       1570         STA  212       ; TELL BASIC WHAT'S WRONG
06A9 A900       1580         LDA  #0
06AB 85D5       1590         STA  213
                1600 ;NOW FALL THROUGH TO CLOSEIT
                1610 ;THEN RETURN TO BASIC
06AD A90C       1620 CLOSEIT LDA #CLOSE
06AF 9D4203     1630         STA  ICCOM,X
06B2 2056E4     1640         JSR  CIO       ; CLOSE FILE
06B5 60         1650         RTS
06B6            1660         .END
```

LOADIT.OBJ
SUBROUTINES

=0010 IOCB1	=0340 ICHID	=0341 ICDNO	=0342 ICCOM
=0343 ICSTA	=0344 ICBAL	=0346 ICPTL	=0348 ICBLL
=034A ICAX1	=034B ICAX2	=034C ICAX3	=034D ICAX4
=034E ICAX5	=034F ICAX6	=E456 CIO	=0003 ENR
=0088 EOF	=0003 OPEN	=0007 GETCHR	=000C CLOSE
=0004 OREAD	=02E0 RUNLOC	=00CB FREEO	=00CB HEADER
=00CF FLAG	0600 INIT	06AD CLOSEIT	0622 RDHDR
06A6 ERROR	0646 CHKID	06A0 DONE	0650 CONT
0661 HDROK	0673 RDBLOK		

```
10 ;ATARI 800 SCREEN DUMP UTILITY
20 ;FOR DUMPING GRAPHICS 8 PICTURES
30 ;TO EPSON MX-100 PRINTER.
40 ;WRITTEN BY DAVID NEWCORN 2/28/82
50 ;ASSEMBLY LANGUAGE LISTING
60 ;IN COMPUTE!'S SECOND BOOK OF ATARI
70 ;
80 ;PAGES 11-14
90 ; ASM ,#P,#D:SCRDUMP.OBJ
```

PROGRAM LISTING

```
0000          0100          .PAGE "PROGRAM LISTING"
FFFF          0110 ADR      =     $FFFF      ;DUMMY ADDRESS (SCREEN MEM ADDR)
FFFF          0120 STR      =     $FFFF      ;DUMMY ADDRESS (STRING ADDR)
              0130 ;
0000          0140          *=    $0600
0600 68       0150          PLA              ; PULL OFF AUX BYTE FROM BASIC CALL
              0160 ;
0601 68       0170          PLA              ; PULL HIGH BYTE OF STRING STORED
0602 8D1506   0180          STA   LOA+2      ; STORE HIGH BYTE
0605 68       0190          PLA              ; PULL LOW BYTE
0606 8D1406   0200          STA   LOA+1      ; THEN SAVE IT
0609 68       0210          PLA              ;PULL HIGH BYTE OF BEGINNING OF SCREEN M
060A 8D1B06   0220          STA   STO+2
060D 68       0230          PLA              ; THE LOW BYTE
060E 8D1A06   0240          STA   STO+1
0611 A0C1     0250          LDY   #193       ; LOAD Y AXIS COUNTER
              0260 ;
0613 ADFFFF   0270 LOA      LDA   ADR        ; LOAD SCREEN BYTE
0616 88       0280          DEY              ; DECREMENT COUNTER
0617 F023     0290          BEQ   RET        ; RETURN TO BASIC IF DONE
              0300 STO
0619 8DFFFF   0310          STA   STR        ; STORE SCREEN BYTE IN A$
061C EE1A06   0320          INC   STO+1      ; INCREMENT LOW END OF STRING
061F F015     0330          BEQ   BIG1       ; IF LOW END OVERFLOWS, THEN INC HI END
              0340 CONT
0621 AD1406   0350          LDA   LOA+1      ; LOAD ACCUM WITH LOW SCREEN ADDRESS
0624 38       0360          SEC              ; SUBTRACT WITHOUT BORROW
0625 E928     0370          SBC   #40        ; SUBTRACT 40 (40 BYTES PER SCAN LINE).
0627 8D1406   0380          STA   LOA+1      ; STORE RESULT
062A 9004     0390          BCC   BIG        ; IF UNDERFLOW, DEC HIGH BYTE OF SCREEN
062C 18       0400          CLC
062D 4C1306   0410          JMP   LOA
              0420 BIG
0630 CE1506   0430          DEC   LOA+2      ; DECREMENT HIGH BYTE OF SCREEN MEM
0633 4C1306   0440          JMP   LOA        ; LOAD NEXT BYTE.
              0450 ;
              0460 BIG1
0636 EE1B06   0470          INC   STO+2      ; INCREMENT HIGH BYTE OF STRING STORAGE
0639 4C2106   0480          JMP   CONT       ; CONTINUE
              0490 ;
063C 60       0500 RET      RTS
063D          0510          .END
```

PROGRAM LISTING

=FFFF ADR	=FFFF STR	0613 LOA	0619 STO
063C RET	0636 BIG1	0621 CONT	0630 BIG

GRAPHICS ROUTINES COMPUTE! #21

```
0000          10          .PAGE "GRAPHICS ROUTINES COMPUTE! #21"
              20 ;FROM COMPUTE #21: GRAPHICS ROUTINES
              30 ;
              40 ; ASM,#P,#D:GRAPHICS.OBJ
              50 ; LIST#D:GRAPHICS.SRC
              60 ; CIO EQUATES
              70 ;
E456          80 CIO      =     $E456      ; CALL OS THRU HERE
0342          90 ICCOM    =     $342       ; COMMAND TO CIO IN IOCB
0344         0100 ICBADR  =     $344       ; BUFFER OR FILENAME ADDRESS
0348         0110 ICBLEN  =     $348       ; BUFFER LENGTH
034A         0120 ICAUX1  =     $34A       ; AUXILLIARY BYTE #1
034B         0130 ICAUX2  =     $34B       ; AUXIALLARY BYTE #2
0003         0140 COPN    =     3          ; COMMAND OPEN
000C         0150 CCLOSE  =     12         ; COMMAND CLOSE
0007         0160 CGBINR  =     7          ; COMMAND GET BINARY RECORD
000B         0170 CPBINR  =     11         ; COMMAND PUT BINARY RECORD
0011         0180 CDRAW   =     17         ; COMMAND DRAWTO
0012         0190 CFILL   =     18         ; COMMAND FILL (NOT USED IN THIS DEMO.)
0004         0200 OPIN    =     4          ; OPEN FOR INPUT
0008         0210 OPOUT   =     8          ; OPEN FOR OUTPUT
             0220 ;
             0230 ; EQUATES USED BY THE "S:" DRIVER AND
             0240 ;         THE VBLANK ROUTINES
             0250 ;
0055         0260 HORIZONTAL = $55
0054         0270 VERTICAL =  $54
02FB         0280 DRAWCOLOR = $2FB
02C4         0290 COLORO =    $2C4
             0300 ;
00FF         0310 LOW     =     $FF
0100         0320 HIGH    =     $100
             0330 RANDOM =    $D20A
             0340 ;
             0350 ;THE ACTUAL ROUTINES
             0360 ;
             0370 ;
             0380 ;==========================================================
```

```
0000          0390          .PAGE "GRAPHICS ROUTINES"
              0400 ;
              0410 ;FIRST, SET THE LOCATION AND SOME MISC. RAM USAGE
              0420 ;
0000          0430          *=   $660
0660 00       0440 SAVECOLOR .BYTE 0          ;WHERE COLOR IS SAVED
0661 53       0450 SNAME    .BYTE "S:",0      ;THE FILENAME FOR OPEN
0662 3A
0663 00
              0460 ;
              0470 ;
              0480 ;GRAPHICS G
              0490 ;
              0500 ;          ENTRY: A-REG CONTAINS GRAPHICS MODE "G"
              0510 ;          EXIT:  Y-REG HAS COMPLETION STATUS
              0520 ;
              0530 GRAPHICS
0664 48       0540          PHA               ; SAVE "G"
0665 A260     0550          LDX  #6*$10       ;FILE 6
0667 A90C     0560          LDA  #CCLOSE
0669 9D4203   0570          STA  ICCOM,X
066C 2056E4   0580          JSR  CIO          ;FIRST, WE MUST CLOSE FILE #6
              0590 ;(WE IGNORE ANY ERRORS FROM THE CLOSE)
              0600 ;
066F A260     0610          LDX  #6*$10       ; AGAIN, FILE 6
0671 A903     0620          LDA  #CDPN        ; WE WILL OPEN THIS "FILE"
0673 9D4203   0630          STA  ICCOM,X
0676 A961     0640          LDA  #SNAME&LOW
0678 9D4403   0650          STA  ICBADR,X     ; WE USE THE FILE NAM "S:"
067B A906     0660          LDA  #SNAME/HIGH
067D 9D4503   0670          STA  ICBADR+1,X   ; BY POINTING TO IT.
              0680 ;
              0690 ;ALL IS SET UP FOR OPEN, NOW
              0700 ;WE TELL CIO (AND S:) WHAT KIND OF OPEN
              0710 ;
0680 68       0720          PLA               ; OUR SAVED "G" GRAPHICS MODE
0681 9D4B03   0730          STA  ICAUX2,X     ; IS GIVEN TO "S:"
              0740 ;
              0750 ;  (NOTE THAT S: IGNORES THE UPPER BITS OF AUX2)
              0760 ;
0684 29F0     0770          AND  #$F0         ; NOW WE GET JUST THE UPPER BITS
0686 4910     0780          EOR  #$10         ; AND FLIP BIT 4
              0790 ;
              0800 ;(READ THE TEXT. S: EXPECTS THIS BIT INVERTED
              0810 ;          FROM WHAT NORMAL BASIC USAGE IS.)
              0820 ;
0688 090C     0830          ORA  #$0C         ; ALLOW READ AND WRITE ACCESS (FOR CIO)
068A 9D4A03   0840          STA  ICAUX1,X     ; MAKE CIO AND S: HAPPY
068D 2056E4   0850          JSR  CIO          ;    AND DO THE OPEN OF S:
0690 60       0860          RTS
              0870 ;=========================================================
```

```
                    0880 ;
                    0890 ;          COLOR C
                    0900 ;
                    0910 ;
                    0920 ;         ENTER: COLOR "C" IN A-REGISTER
                    0930 ;          EXIT: UNCHANGED
                    0940 ;
                    0950 COLOR
0691 8D6006         0960          STA  SAVECOLOR
0694 60             0970          RTS             ; EXCITING, WASN'T IT?
                    0980 ;
                    0990 ;=========================================================
                    1000 ;
                    1010 ;    POSITION H,V
                    1020 ;
                    1030 ;         ENTER: H (HORIZ) POSITION IN X,A REGISTERS (LSB,MSB)
                    1040 ;                V (VERT) POSITION IN Y-REGISTER
                    1050 ;
                    1060 ;          EXIT: UNCHANGED
                    1070 ;
                    1080 POSITION
0695 8655           1090          STX  HORIZONTAL
0697 8556           1100          STA  HORIZONTAL+1 ; READ THE TEXT (MAGAZINE)
0699 8454           1110          STY  VERTICAL  ; TOO SIMPLE, RIGHT?
069B 60             1120          RTS
                    1130 ;
                    1140 ;=========================================================
                    1150 ;
                    1160 ; PLOT H,V
                    1170 ;
                    1180 ;         ENTER: MUST HAVE DONE A PREVIOUS COLOR CALL
                    1190 ;                X,A, AND Y REGISTERS SET AS IN POSITION
                    1200 ;
                    1210 ;
                    1220 PLOT
069C 209506         1230          JSR  POSITION
069F A260           1240          LDX  #6*$10     ; FILE 6, AGAIN
06A1 A90B           1250          LDA  #CPBINR    ; COMMAND PUT BINARY RECORD
06A3 9D4203         1260          STA  ICCOM,X
06A6 A900           1270          LDA  #0
06A8 9D4803         1280          STA  ICBLEN,X
06AB 9D4903         1290          STA  ICBLEN+1,X ; IF BUFFER LENGTH IS ZERO...
06AE AD6006         1300          LDA  SAVECOLOR ; THEN CPBINR PUTS ONE CHAR FROM A-REG
06B1 2056E4         1310          JSR  CIO        ; AND THIS IS HOW WE PLOT
06B4 60             1320          RTS
                    1330 ;
                    1340 ;=========================================================
                    1350 ;
                    1360 ; LOCATE H,V,C
                    1370 ;
                    1380 ;         ENTER: X,A, AND Y REGISTERS SET UP AS IN POSITION
                    1390 ;          EXIT: A-REG HAS THE LOCATED COLOR
```

```
                1400 ;                  Y-REG HAS THE COMPLETION CODE
                1410 ;
                1420 LOCATE
06B5 209506     1430            JSR  POSITION
06B8 A260       1440            LDX  #6*$10
06BA A907       1450            LDA  #CGBINR    ; COMMAND GET BINARY RECORD
06BC 9D4203     1460            STA  ICCOM,X
06BF A900       1470            LDA  #0
06C1 9D4803     1480            STA  ICBLEN,X
06C4 9D4903     1490            STA  ICBLEN+1,X ; IF BUFFER LENGTH IS ZERO,
06C7 2056E4     1500            JSR  CIO        ; THEN THE CHAR IS RETURNED IN A
06CA 60         1510            RTS
                1520 ;
                1530 ;=========================================================
                1540 ;
                1550 ;          DRAWTO H,V
                1560 ;
                1570 ;          ENTER: MUST HAVE DONE A PREVIOUS PLOT
                1580 ;                   X,A, AND Y REGISTERS AS IN POSITION
                1590 ;
                1600 ;           EXIT: Y-REGISTER HAS COMPLETION CODE
                1610 ;
                1620 ;
                1630 DRAWTO
06CB 209506     1640            JSR  POSITION
06CE AD6006     1650            LDA  SAVECOLOR
06D1 8DFB02     1660            STA  DRAWCOLOR ; WHERE DRAWTO EXPECTS ITS COLOR
06D4 A260       1670            LDX  #6*$10    ; FILE #6
06D6 A911       1680            LDA  #CDRAW    ; JUST A COMMAND TO "S:"
06D8 9D4203     1690            STA  ICCOM,X
06DB A90C       1700            LDA  #$0C
06DD 9D4A03     1710            STA  ICAUX1,X  ; INSURANCE
06E0 A900       1720            LDA  #0
06E2 9D4B03     1730            STA  ICAUX2,X  ; ...GUARANTEED TO WORK
06E5 2056E4     1740            JSR  CIO       ; DO THE ACTUAL DRAWTO
06E8 60         1750            RTS
                1760 ;
                1770 ;=========================================================
                1780 ;
                1790 ;  SETCOLOR REG,HUE,LUM
                1800 ;
                1810 ;          ENTER: X-REG HAS COLOR REGISTER
                1820 ;                 A-REG HAS HUE
                1830 ;                 Y-REG HAS LUMINANCE
                1840 ;
                1850 ;           EXIT: (UNDEFINED)
                1860 ;
                1870 SETCOLOR
06E9 0A         1880            ASL  A
06EA 0A         1890            ASL  A
06EB 0A         1900            ASL  A
06EC 0A         1910            ASL  A          ; WE NEED HUE * 16
```

GRAPHICS ROUTINES

```
06ED 9DC402 1920          STA  COLORO,X  ; SAVE IT HERE FOR A WHILE
06F0 98     1930          TYA
06F1 290E   1940          AND  #$0E      ; ONLY LUMINANCE BITS THAT MATTER
06F3 18     1950          CLC
06F4 7DC402 1960          ADC  COLORO,X  ; END OF THE WHILE
06F7 9DC402 1970          STA  COLORO,X  ; AND VBLANK WILL MOVE THIS TO HARDWARE
06FA 60     1980          RTS
            1990 ;
            2000 ;=========================================================
            2010 ;
            2020 ; RND R
            2030 ;
            2040 ;        ENTER: R IN ACCUMULATOR
            2050 ;         EXIT: A-REGISTER CONTAINS A RANDOM
            2060 ;                NUMBER FROM 0 TO R, INCLUSIVE
            2070 ;
            2080 RND
06FB 8D0A07 2090          STA  RTEMP
06FE EE0A07 2100          INC  RTEMP     ; MAKES CMP EASIER
            2110 RNDWAIT
0701 AD0000 2120          LDA  RANDOM    ; SET A RANDOM NUMBER
)***ERROR - 5-UNDEFINED
0704 CD0A07 2130          CMP  RTEMP     ; TOO BIG?
0707 B0F8   2140          BCS  RNDWAIT   ; YES...WAIT
0709 60     2150          RTS
            2160 ;
070A 00     2170 RTEMP    .BYTE 0        ; RESERVE SPACE
            2180 ;
            2190 ;=========================================================
            2200 ;
070B        2210          .END
```

GRAPHICS ROUTINES

```
=E456 CIO          =0342 ICCOM        =0344 ICBADR       =0348 ICBLEN
=034A ICAUX1       =034B ICAUX2       =0003 COPN         =000C CCLOSE
=0007 CGBINR       =000B CPBINR       =0011 CDRAW        =0012 CFILL
=0004 OPIN         =0008 OPOUT        =0055 HORIZONTAL   =0054 VERTICAL
=02FB DRAWCOLOR    =02C4 COLORQ       =00FF LOW          =0100 HIGH
 0660 SAVECOLOR     0661 SNAME         0664 GRAPHICS      0691 COLOR
 0695 POSITION      069C PLOT          06B5 LOCATE        06CB DRAWTO
 06E9 SETCOLOR      06FB RND           070A RTEMP         0701 RNDWAIT
?0000 RANDOM
```

```
1 REM SAVE"D:GTIA.11
10 R=16:X=0:C=15
20 GRAPHICS 9:SETCOLOR 4,13,0
30 FOR I=130 TO 191
40   COLOR C
50   PLOT 0,I:DRAWTO 79,I
60   X=X+1:IF X=4 THEN X=0:C=C-1
70   NEXT I
80 FOR I=0 TO 79 STEP 8
90   COLOR 3:PLOT 59,130:DRAWTO I,191
100   NEXT I
110 COLOR 1:FOR I=0 TO 7:PLOT 2,164:DRAWTO 21,158+I:NEXT I
120 COLOR 15:FOR I=0 TO 3:PLOT 21,140:DRAWTO 21+I,164-I*2:NEXT I
130 COLOR 4:FOR I=0 TO 4:PLOT 20,140:DRAWTO 17+I,160+I:NEXT I
140 FOR Z=1 TO 15
150   FOR I=0 TO 360 STEP 6
160     X=0.25*R*COS(I)+35
170     Y=R*SIN(I)+50
180     COLOR Z:PLOT X,Y
200     PLOT X+10,Y+17:PLOT X+30,Y-20:PLOT X-2,Y+12:PLOT X+21,Y+70
240     NEXT I
250   R=R-1
260   NEXT Z
270 FOR I=2 TO 4:COLOR I:PLOT 46,72:DRAWTO 51+I,106
280   PLOT 43,62:DRAWTO 39,50+I
290   PLOT 47,62:DRAWTO 60+I,35
300   NEXT I
310 GOTO 310
```

```
10 REM SAVE"D:BOING.BAP"
20 REM COMPUTE #27
30 REM BY BILL WILKINSON
40 REM
100 GRAPHICS 3
200 XMOVE=INT(5*RND(0))-2
300 YMOVE=INT(5*RND(0))-2
400 IF XMOVE+YMOVE=0 THEN 200
500 X=INT(40*RND(0))
600 Y=INT(20*RND(0))
700 XNEW=X:YNEW=Y
900 POKE 19,0:POKE 20,0:REM = RESET TIMER
1000 REM LOOP STARTS HERE
1100 COLOR 0:PLOT X,Y
1200 COLOR 2:PLOT XNEW,YNEW
1300 X=XNEW:Y=YNEW
1400 XNEW=X+XMOVE:YNEW=Y+YMOVE
1500 IF XNEW<=0 OR XNEW>=39 THEN XMOVE=-XMOVE
1600 IF XNEW<0 OR XNEW>39 THEN XNEW=X
1700 IF YNEW<=0 OR YNEW>=19 THEN YMOVE=-YMOVE
1800 IF YNEW<0 OR YNEW>19 THEN YNEW=Y
1900 IF PEEK(19)=0 THEN 1000
2000 RUN
```

```
1000 REM SAVE"D:BOING3.BAP
1010 REM INSIGHT ATARI - B. WILKINSON
1020 REM
1030 REM COMPUTE! - SEPT. 1982 (#28)
1040 REM
1050 DIM YP(1),SCORE(1):SCORE(0)=0:SCORE(1)=0:SINGLE=(PEEK(53279)<>7)
1060 LASTWIN=1:IF RND(0)>=0.5 THEN LASTWIN=-LASTWIN
2000 REM ****  PREPARE FOR A SERVE  ****
2010 GRAPHICS 3:COLOR 2:PLOT 0,0:DRAWTO 39,0
2020 PLOT 0,19:DRAWTO 39,19
2030 PRINT :PRINT SCORE(1),,SCORE(0):PRINT "   -     SCORE";
2035 IF SCORE(0)>20 OR SCORE(1)>20 THEN END
2040 COLOR 3:PLOT 0,9:DRAWTO 0,11:PLOT 39,9:DRAWTO 39,11
2050 IF SINGLE THEN COLOR 2:PLOT 39,0:DRAWTO 39,19
2060 YP(0)=10:YP(1)=10:REM VERTICAL POSITION
2070 IF SINGLE THEN LASTWIN=1
2100 REM *****   SETUP BALL  ******
2110 XMOVE=LASTWIN:YMOVE=INT(3*RND(0))-1:Y=INT(12*RND(0))+4
2120 YNEW=Y:X=19-5*XMOVE:XNEW=X
2500 REM ***  MAIN PLAYING LOOP ***
2510 REM
2520 REM 1. CHECK AND MOVE PADDLES
2530 REM 2. SHOW NEW BALL POSITION
2540 REM 3. CHECK FOR COLLISIONS, ETC.
2550 REM
2590 REM ***  FIRST CHECK AND MOVE PADDLES ***
2595 REM
2600 V0=PTRIG(0)-PTRIG(1):IF  NOT V0 THEN 2700
2610 VP0=YP(0)-V0:IF VP0<2 OR VP0>17 THEN 2700
2620 COLOR 0:PLOT 0,YP(0)+V0:COLOR 3:PLOT 0,VP0-V0:YP(0)=VP0
2700 V1=PTRIG(2)-PTRIG(3):IF SINGLE OR V1=0 THEN 3000
2710 VP1=YP(1)-V1:IF VP1<2 OR VP1>17 THEN 3000
2720 COLOR 0:PLOT 39,YP(1)+V1:COLOR 3:PLOT 39,VP1-V1:YP(1)=VP1
3000 REM *** BALL CONTROL ***
3010 COLOR 0:PLOT X,Y
3014 IF XNEW<0 THEN XNEW=0:XMOVE=-XMOVE
3015 IF YNEW<1 THEN YNEW=1:YMOVE=-YMOVE
3016 IF YNEW>18 THEN YNEW=18:YMOVE=-YMOVE
3020 COLOR 1:PLOT XNEW,YNEW
3030 X=XNEW:Y=YNEW
3040 XNEW=XNEW+XMOVE:YNEW=YNEW+YMOVE
3050 IF XNEW<38 AND XNEW>1 THEN 3200
3060 HITP=(XNEW>20):XHIT=39*HITP
3070 IF SINGLE THEN IF HITP THEN 3100
3080 YMSAVE=YMOVE:YNEW=INT(YNEW):YMOVE=(YNEW-YP(HITP))/2
3090 IF ABS(YMOVE)>1 THEN 4000
3100 XMOVE=-XMOVE
3200 IF YNEW=1 OR YNEW=18 THEN YMOVE=-YMOVE
3290 GOTO 2600
4000 REM ***  THE "LOSE" ROUTINE   ******
4010 COLOR 0:PLOT X,Y
4020 COLOR 1:PLOT XNEW,YNEW
4030 FOR I=1 TO 10:NEXT I
4040 COLOR 0:PLOT XNEW,YNEW
4045 IF YMSAVE<-1 THEN YMSAVE=1
4046 IF YMSAVE>19 THEN YMSAVE=18
4050 COLOR 2:PLOT XNEW+XMOVE,YNEW+YMSAVE
4130 SOUND 0,132,12,12:POKE 20,0
4140 SETCOLOR 1,0,PEEK(20)*4:IF PEEK(20)<32 THEN 4140
4150 SOUND 0,0,0,0
4200 REM ****   SCORE IT   ********
4210 SCORE(HITP)=SCORE(HITP)+1
4220 LASTWIN=1:IF HITP THEN LASTWIN=-LASTWIN
4990 GOTO 2000
```

```
INITIALIZATION

0000            20          .PAGE "INITIALIZATION"
                30 ;
                40 ; A SIMPLE BOUNCING BALL PROGRAM
                50 ;
                60 ; LIST#D:BOING.SRC
                70 ; ASM ,#P:,#D:BOING.OBJ
```

```
BOING PROGRAM W/BASIC LINES

0000          80          .PAGE "BOING PROGRAM W/BASIC LINES"
0000          90          *=    $3000
              0100 LINE100
              0110 ;>>>
 GRAPHICS 3
3000 A903     0120          LDA   #3
3002 20E830   0130          JSR   GRAPHICS
              0140 ;
              0150 LINE200
              0160 ;>>>
 XMOVE=INT(5*RND(0))-2
3005 A904     0170          LDA   #4
3007 207F31   0180          JSR   RND         ; GET RANDOM NUM FROM 0 TO 4
300A 38       0190          SEC
300B E902     0200          SBC   #2          ; NOW IS RANDOM FROM -2 TO +2
300D 8DE230   0210          STA   XMOVE       ; AS IN BASIC PROGRAM
              0220 ;
              0230 LINE300
              0240 ;>>>
 YMOVE=INT(5*RND(0))-2
3010 A904     0250          LDA   #4
3012 207F31   0260          JSR   RND
3015 38       0270          SEC
3016 E902     0280          SBC   #2
3018 8DE330   0290          STA   YMOVE
              0300 LINE400
              0310 ;>>>
 IF XMOVE+YMOVE=0 THEN 200
301B ADE230   0320          LDA   XMOVE
301E 18       0330          CLC
301F 6DE330   0340          ADC   YMOVE
3022 F0E1     0350          BEQ   LINE200
              0360 LINE500
              0370 ;>>>
 X=INT(40*RND(0))
              0380 ;
3024 A927     0390          LDA   #39
3026 207F31   0400          JSR   RND
3029 8DDE30   0410          STA   X           ; STORE THE RANDOM #
              0420 LINE600
              0430 ;>>>
 Y=INT(20*RND(0))
              0440 ;
302C A913     0450          LDA   #19
302E 207F31   0460          JSR   RND
3031 8DDF30   0470          STA   Y
              0480 ;
              0490 LINE700
              0500 ;>>>
 XNEW=X : YNEW=Y
              0510 ;
3034 ADDE30   0520          LDA   X
3037 8DE030   0530          STA   XNEW
303A ADDF30   0540          LDA   Y
303D 8DE130   0550          STA   YNEW
              0560 ;
              0570 LINE900
```

```
BOING PROGRAM W/BASIC LINES

              0590 ;>>>
 POKE 19,0 : POKE 20,0
              0600 ;
3040 A900     0610         LDA  #0
3042 8513     0620         STA  19
3044 8514     0630         STA  20
              0640 ;
              0650 LINE1000
              0660 ;
              0670 ;>>>
 REM LOOP STARTS HERE
              0680 ;
              0690 LINE1100
              0700 ;>>>
 COLOR 0: PLOT X,Y
              0710 ;
3046 A900     0720         LDA  #0
3048 201531   0730         JSR  COLOR
304B AEDE30   0740         LDX  X
304E ACDF30   0750         LDY  Y          ; LOAD VALUES FOR SUBROUTINES
3051 202031   0760         JSR  PLOT
              0770 ;
              0780 LINE1200
              0790 ;>>>
 COLOR 2: PLOT XNEW,YNEW
3054 A902     0800         LDA  #2
3056 201531   0810         JSR  COLOR
3059 A900     0820         LDA  #0         ; (NEEDED FOR PLOT)
305B AEE030   0830         LDX  XNEW
305E ACE130   0840         LDY  YNEW
3061 202031   0850         JSR  PLOT
              0860 ;
              0870 LINE1300
              0880 ;
              0890 ;>>>
  X=XNEW : Y=YNEW
3064 ADE030   0900         LDA  XNEW
3067 8DDE30   0910         STA  X
306A ADE130   0920         LDA  YNEW
306D 8DDF30   0930         STA  Y
              0940 ;
              0950 LINE1400
              0960 ;
              0970 ;>>>
 XNEW = X + XMOVE : YNEW = Y + YMOVE
              0980 ;
3070 ADDE30   0990         LDA  X
3073 18       1000         CLC
3074 6DE230   1010         ADC  XMOVE
3077 8DE030   1020         STA  XNEW
307A ADDF30   1030         LDA  Y
307D 18       1040         CLC
307E 6DE330   1050         ADC  YMOVE
3081 8DE130   1060         STA  YNEW
              1070 ;
              1080 LINE1500
              1090 ;>>>
```

```
IF XNEW <= 0 OR XNEW >= 39 THEN XMOVE = -XMOVE
3084 ADE030 1100          LDA  XNEW
```

```
BOING PROGRAM W/BASIC LINES

3087 3006    1110          BMI   THEN1500   ; XNEW < 0
3089 F004    1120          BEQ   THEN1500   ; XNEW = 0
308B C927    1130          CMP   #39
308D 9009    1140          BCC   LINE1600   ; XNEW NOT >= 39
             1150 THEN1500
308F A900    1160          LDA   #0
3091 38      1170          SEC
3092 EDE230  1180          SBC   XMOVE      ; GET 0 - XMOVE
3095 8DE230  1190          STA   XMOVE      ;  TO XMOVE
             1200 ;
             1210 ;
             1220 LINE1600
             1230 ;>>>
 IF XNEW<0 OR XNEW>39 THEN XNEW=X
             1240 ;
3098 ADE030  1250          LDA   XNEW
309B 3006    1260          BMI   THEN1600   ; XNEW < 0
309D C927    1270          CMP   #39
309F F008    1280          BEQ   LINE1700   ; XNEW = 39
30A1 9006    1290          BCC   LINE1700
             1300 THEN1600
30A3 ADDE30  1310          LDA   X
30A6 8DE030  1320          STA   XNEW       ; XNEW = X
             1330 ;
             1340 LINE1700
             1350 ;>>>
 IF YNEW<=0 OR YNEW>=19 THEN YMOVE= -YMOVE
             1360 ;
30A9 ADE130  1370          LDA   YNEW
30AC 3006    1380          BMI   THEN1700   ; YNEW < 0
30AE F004    1390          BEQ   THEN1700   ; YNEW = 0
30B0 C913    1400          CMP   #19
30B2 9009    1410          BCC   LINE1800   ; YNEW NOT >= 19
             1420 THEN1700
30B4 A900    1430          LDA   #0
30B6 38      1440          SEC
30B7 EDE330  1450          SBC   YMOVE
30BA 8DE330  1460          STA   YMOVE      ; YMOVE = -YMOVE
             1470 ;
             1480 LINE1800
             1490 ;>>>
 IF YNEW < 0 OR YNEW > 39 THEN YNEW = Y
             1500 ;
30BD ADE130  1510          LDA   YNEW
30C0 3006    1520          BMI   THEN1800
30C2 C913    1530          CMP   #19
30C4 F008    1540          BEQ   LINE1900
30C6 9006    1550          BCC   LINE1900
             1560 THEN1800
30C8 ADDF30  1570          LDA   Y
30CB 8DE130  1580          STA   YNEW       ; YNEW = Y
             1590 ;
             1600 LINE1900
             1610 ;>>>
 IF PEEK(19)=0 THEN 1000
             1620 ;
```

```
BOING PROGRAM W/BASIC LINES

30CE A513    1630          LDA  19
30D0 D009    1640          BNE  LINE2000
             1650 ;=== LEAVE THESE 6 LINES OUT FIRST TIME ===
30D2 A514    1660          LDA  20          ; LSB OF CLOCK
             1670 CLOCKWAIT
30D4 C514    1680          CMP  20          ; CHANGED YET?
30D6 F0FC    1690          BEQ  CLOCKWAIT   ; NO...WAIT
             1700 ;=== BUT ALWAYS KEEP LINE 1710 ===
30D8 4C4630  1710          JMP  LINE1000
             1720 ;
             1730 LINE2000
             1740 ;>>>
 RUN
             1750 ;
30DB 4C0030  1760          JMP  LINE100
```

```
30DE          1770          .PAGE "VARIABLES AND SUBROUTINES"
              1780 ;
              1790 ; THERE ARE NO DIRECT BASIC EQUIVALENTS
              1800 ; TO THE FOLLOWING LINES ... THE BASIC
              1810 ; INTERPRETER HANDLES ALL THIS FOR YOU
              1820 ; AUTOMATICALLY.
              1830 ;
              1840 ; FIRST --- THE VARIABLES:
              1850 ;
30DE 00       1860 X        .BYTE 0
30DF 00       1870 Y        .BYTE 0
30E0 00       1880 XNEW     .BYTE 0
30E1 00       1890 YNEW     .BYTE 0
30E2 00       1900 XMOVE    .BYTE 0
30E3 00       1910 YMOVE    .BYTE 0
              1920 ;
              1930 ;
              1940 ; SECOND ---- THE SUBROUTINES:
              1950 ;
              1960 ; MOSTLY FROM COMPUTE! OF FEBRUARY 1982
              1970 ; ISSUE #21. RND HAS BEEN ADDED
              1980 ;
30E4          1990          .INCLUDE #D:GRAPHICS.SRC
```

```
30E4          10          .PAGE "GRAPHICS ROUTINES COMPUTE! #21"
              20 ;FROM COMPUTE #21: GRAPHICS ROUTINES
              30 ;
              40 ; ASM,#P,#D:GRAPHICS.OBJ
              50 ; LIST#D:GRAPHICS.SRC
              60 ; CIO EQUATES
              70 ;
E456          80 CIO      =    $E456      ; CALL OS THRU HERE
0342          90 ICCOM    =    $342       ; COMMAND TO CIO IN IOCB
0344         0100 ICBADR  =    $344       ; BUFFER OR FILENAME ADDRESS
0348         0110 ICBLEN  =    $348       ; BUFFER LENGTH
034A         0120 ICAUX1  =    $34A       ; AUXILLIARY BYTE #1
034B         0130 ICAUX2  =    $34B       ; AUXIALLARY BYTE #2
0003         0140 COPN    =    3          ; COMMAND OPEN
000C         0150 CCLOSE  =    12         ; COMMAND CLOSE
0007         0160 CGBINR  =    7          ; COMMAND GET BINARY RECORD
000B         0170 CPBINR  =    11         ; COMMAND PUT BINARY RECORD
0011         0180 CDRAW   =    17         ; COMMAND DRAWTO
0012         0190 CFILL   =    18         ; COMMAND FILL (NOT USED IN THIS DEMO.)
0004         0200 OPIN    =    4          ; OPEN FOR INPUT
0008         0210 OPOUT   =    8          ; OPEN FOR OUTPUT
             0220 ;
             0230 ; EQUATES USED BY THE "S:" DRIVER AND
             0240 ;         THE VBLANK ROUTINES
             0250 ;
0055         0260 HORIZONTAL = $55
0054         0270 VERTICAL =  $54
02FB         0280 DRAWCOLOR = $2FB
02C4         0290 COLOR0 =    $2C4
             0300 ;
00FF         0310 LOW     =    $FF
0100         0320 HIGH    =    $100
D20A         0330 RANDOM  =    $D20A
             0340 ;
             0350 ;THE ACTUAL ROUTINES
             0360 ;
             0370 ;
             0380 ;=========================================================
```

```
30E4            0390          .PAGE "GRAPHICS ROUTINES"
                0400 ;
                0410 ;FIRST, SET THE LOCATION AND SOME MISC. RAM USAGE
                0420 ;
30E4 00         0440 SAVECOLOR .BYTE 0        ;WHERE COLOR IS SAVED
30E5 53         0450 SNAME    .BYTE "S:",0    ;THE FILENAME FOR OPEN
30E6 3A
30E7 00
                0460 ;
                0470 ;
                0480 ;GRAPHICS G
                0490 ;
                0500 ;          ENTRY: A-REG CONTAINS GRAPHICS MODE "G"
                0510 ;          EXIT:  Y-REG HAS COMPLETION STATUS
                0520 ;
                0530 GRAPHICS
30E8 48         0540          PHA             ; SAVE "G"
30E9 A260       0550          LDX  #6*$10     ;FILE 6
30EB A90C       0560          LDA  #CCLOSE
30ED 9D4203     0570          STA  ICCOM,X
30F0 2056E4     0580          JSR  CIO        ;FIRST, WE MUST CLOSE FILE #6
                0590 ;(WE IGNORE ANY ERRORS FROM THE CLOSE)
                0600 ;
30F3 A260       0610          LDX  #6*$10     ; AGAIN, FILE 6
30F5 A903       0620          LDA  #COPN      ; WE WILL OPEN THIS "FILE"
30F7 9D4203     0630          STA  ICCOM,X
30FA A9E5       0640          LDA  #SNAME&LOW
30FC 9D4403     0650          STA  ICBADR,X   ; WE USE THE FILE NAM "S:"
30FF A930       0660          LDA  #SNAME/HIGH
3101 9D4503     0670          STA  ICBADR+1,X ; BY POINTING TO IT.
                0680 ;
                0690 ;ALL IS SET UP FOR OPEN, NOW
                0700 ;WE TELL CIO (AND S:) WHAT KIND OF OPEN
                0710 ;
3104 68         0720          PLA             ; OUR SAVED "G" GRAPHICS MODE
3105 9D4B03     0730          STA  ICAUX2,X   ; IS GIVEN TO "S:"
                0740 ;
                0750 ;  (NOTE THAT S: IGNORES THE UPPER BITS OF AUX2)
                0760 ;
3108 29F0       0770          AND  #$F0       ; NOW WE GET JUST THE UPPER BITS
310A 4910       0780          EOR  #$10       ; AND FLIP BIT 4
                0790 ;
                0800 ;(READ THE TEXT. S: EXPECTS THIS BIT INVERTED
                0810 ;          FROM WHAT NORMAL BASIC USAGE IS.)
                0820 ;
310C 090C       0830          ORA  #$0C       ; ALLOW READ AND WRITE ACCESS (FOR CIO)
310E 9D4A03     0840          STA  ICAUX1,X   ; MAKE CIO AND S: HAPPY
3111 2056E4     0850          JSR  CIO        ;   AND DO THE OPEN OF S:
3114 60         0860          RTS
                0870 ;==========================================================
                0880 ;
```

```
                0890 ;          COLOR C
                0900 ;
                0910 ;
                0920 ;          ENTER: COLOR "C" IN A-REGISTER
                0930 ;           EXIT: UNCHANGED
                0940 ;
                0950 COLOR
3115 8DE430     0960            STA  SAVECOLOR
3118 60         0970            RTS             ; EXCITING, WASN'T IT?
                0980 ;
                0990 ;=========================================================
                1000 ;
                1010 ;    POSITION H,V
                1020 ;
                1030 ;         ENTER: H (HORIZ) POSITION IN X,A REGISTERS (LSB,MSB)
                1040 ;                V (VERT) POSITION IN Y-REGISTER
                1050 ;
                1060 ;          EXIT: UNCHANGED
                1070 ;
                1080 POSITION
3119 8655       1090            STX  HORIZONTAL
311B 8556       1100            STA  HORIZONTAL+1 ; READ THE TEXT (MAGAZINE)
311D 8454       1110            STY  VERTICAL  ; TOO SIMPLE, RIGHT?
311F 60         1120            RTS
                1130 ;
                1140 ;=========================================================
                1150 ;
                1160 ; PLOT H,V
                1170 ;
                1180 ;         ENTER: MUST HAVE DONE A PREVIOUS COLOR CALL
                1190 ;                X,A, AND Y REGISTERS SET AS IN POSITION
                1200 ;
                1210 ;
                1220 PLOT
3120 201931     1230            JSR  POSITION
3123 A260       1240            LDX  #6*$10     ; FILE 6, AGAIN
3125 A90B       1250            LDA  #CPBINR    ; COMMAND PUT BINARY RECORD
3127 9D4203     1260            STA  ICCOM,X
312A A900       1270            LDA  #0
312C 9D4803     1280            STA  ICBLEN,X
312F 9D4903     1290            STA  ICBLEN+1,X ; IF BUFFER LENGTH IS ZERO...
3132 ADE430     1300            LDA  SAVECOLOR ; THEN CPBINR PUTS ONE CHAR FROM A-REG
3135 2056E4     1310            JSR  CIO        ; AND THIS IS HOW WE PLOT
3138 60         1320            RTS
                1330 ;
                1340 ;=========================================================
                1350 ;
                1360 ; LOCATE H,V,C
                1370 ;
                1380 ;         ENTER: X,A, AND Y REGISTERS SET UP AS IN POSITION
                1390 ;          EXIT: A-REG HAS THE LOCATED COLOR
                1400 ;                Y-REG HAS THE COMPLETION CODE
```

```
                1410 ;
                1420 LOCATE
3139 201931     1430         JSR  POSITION
313C A260       1440         LDX  #6*$10
313E A907       1450         LDA  #CGBINR    ; COMMAND GET BINARY RECORD
3140 9D4203     1460         STA  ICCOM,X
3143 A900       1470         LDA  #0
3145 9D4803     1480         STA  ICBLEN,X
3148 9D4903     1490         STA  ICBLEN+1,X ; IF BUFFER LENGTH IS ZERO,
314B 2056E4     1500         JSR  CIO        ; THEN THE CHAR IS RETURNED IN A
314E 60         1510         RTS
                1520 ;
                1530 ;=========================================================
                1540 ;
                1550 ;         DRAWTO H,V
                1560 ;
                1570 ;         ENTER: MUST HAVE DONE A PREVIOUS PLOT
                1580 ;                X,A, AND Y REGISTERS AS IN POSITION
                1590 ;
                1600 ;          EXIT: Y-REGISTER HAS COMPLETION CODE
                1610 ;
                1620 ;
                1630 DRAWTO
314F 201931     1640         JSR  POSITION
3152 ADE430     1650         LDA  SAVECOLOR
3155 8DFB02     1660         STA  DRAWCOLOR ; WHERE DRAWTO EXPECTS ITS COLOR
3158 A260       1670         LDX  #6*$10    ; FILE #6
315A A911       1680         LDA  #CDRAW    ; JUST A COMMAND TO "S:"
315C 9D4203     1690         STA  ICCOM,X
315F A90C       1700         LDA  #$0C
3161 9D4A03     1710         STA  ICAUX1,X  ; INSURANCE
3164 A900       1720         LDA  #0
3166 9D4B03     1730         STA  ICAUX2,X  ; ...GUARANTEED TO WORK
3169 2056E4     1740         JSR  CIO       ; DO THE ACTUAL DRAWTO
316C 60         1750         RTS
                1760 ;
                1770 ;=========================================================
                1780 ;
                1790 ; SETCOLOR REG,HUE,LUM
                1800 ;
                1810 ;         ENTER: X-REG HAS COLOR REGISTER
                1820 ;                A-REG HAS HUE
                1830 ;                Y-REG HAS LUMINANCE
                1840 ;
                1850 ;          EXIT: (UNDEFINED)
                1860 ;
                1870 SETCOLOR
316D 0A         1880         ASL  A
316E 0A         1890         ASL  A
316F 0A         1900         ASL  A
3170 0A         1910         ASL  A          ; WE NEED HUE * 16
3171 9DC402     1920         STA  COLOR0,X   ; SAVE IT HERE FOR A WHILE
```

```
3174 98      1930          TYA
3175 290E    1940          AND  #$OE      ; ONLY LUMINANCE BITS THAT MATTER
3177 18      1950          CLC
3178 7DC402  1960          ADC  COLORO,X  ; END OF THE WHILE
317B 9DC402  1970          STA  COLORO,X  ; AND VBLANK WILL MOVE THIS TO HARDWARE
317E 60      1980          RTS
             1990 ;
             2000 ;=========================================================
             2010 ;
             2020 ; RND R
             2030 ;
             2040 ;        ENTER: R IN ACCUMULATOR
             2050 ;         EXIT: A-REGISTER CONTAINS A RANDOM
             2060 ;                NUMBER FROM O TO R, INCLUSIVE
             2070 ;
             2080 RND
317F 8D8E31  2090          STA  RTEMP
3182 EE8E31  2100          INC  RTEMP     ; MAKES CMP EASIER
             2110 RNDWAIT
3185 AD0AD2  2120          LDA  RANDOM    ; SET A RANDOM NUMBER
3188 CD8E31  2130          CMP  RTEMP     ; TOO BIG?
318B B0F8    2140          BCS  RNDWAIT   ; YES...WAIT
318D 60      2150          RTS
             2160 ;
318E 00      2170 RTEMP    .BYTE O        ; RESERVE SPACE
             2180 ;
             2190 ;=========================================================
             2200 ;
318F         2210          .END
318F         2210          .END
```

GRAPHICS ROUTINES

3000 LINE100	30E8 GRAPHICS	3005 LINE200	317F RND
30E2 XMOVE	3010 LINE300	30E3 YMOVE	301B LINE400
3024 LINE500	30DE X	302C LINE600	30DF Y
3034 LINE700	30E0 XNEW	30E1 YNEW	3040 LINE900
3046 LINE1000	3046 LINE1100	3115 COLOR	3120 PLOT
3054 LINE1200	3064 LINE1300	3070 LINE1400	3084 LINE1500
308F THEN1500	3098 LINE1600	30A3 THEN1600	30A9 LINE1700
30B4 THEN1700	30BD LINE1800	30C8 THEN1800	30CE LINE1900
30DB LINE2000	30D4 CLOCKWAIT	=E456 CIO	=0342 ICCOM
=0344 ICBADR	=0348 ICBLEN	=034A ICAUX1	=034B ICAUX2
=0003 COPN	=000C CCLOSE	=0007 CGBINR	=000B CPBINR
=0011 CDRAW	=0012 CFILL	=0004 OPIN	=0008 OPOUT
=0055 HORIZONTAL	=0054 VERTICAL	=02FB DRAWCOLOR	=02C4 COLORO
=00FF LOW	=0100 HIGH	=D20A RANDOM	30E4 SAVECOLOR
30E5 SNAME	3119 POSITION	3139 LOCATE	314F DRAWTO
316D SETCOLOR	318E RTEMP	3185 RNDWAIT	

```
1000 REM BOING2.BAP
1010 REM INSIGHT ATARI - B. WILKINSON
1020 REM
1030 REM COMPUTE! - SEPT. 1982 (#28)
1040 REM
1050 DIM YP(1),SCORE(1):SCORE(0)=0:SCORE(1)=0:SINGLE=(PEEK(53279)<>7)
1060 LASTWIN=1:IF RND(0)>=0.5 THEN LASTWIN=-LASTWIN
2000 REM ****  PREPARE FOR A SERVE  ****
2010 GRAPHICS 3:COLOR 2:PLOT 0,0:DRAWTO 39,0
2020 PLOT 0,19:DRAWTO 39,19
2030 PRINT :PRINT SCORE(1),,SCORE(0):PRINT "   -     SCORE";
2035 IF SCORE(0)>20 OR SCORE(1)>20 THEN END
2040 COLOR 3:PLOT 0,9:DRAWTO 0,11:PLOT 39,9:DRAWTO 39,11
2050 IF SINGLE THEN COLOR 2:PLOT 39,0:DRAWTO 39,19
2060 YP(0)=10:YP(1)=10:REM VERTICAL POSITION
2070 IF SINGLE THEN LASTWIN=1
2100 REM ******   SETUP BALL  *******
2110 XMOVE=LASTWIN:YMOVE=INT(3*RND(0))-1:Y=INT(12*RND(0))+4
2120 YNEW=Y:X=19-5*XMOVE:XNEW=X
2500 REM ***  MAIN PLAYING LOOP ***
2510 REM
2520 REM 1. CHECK AND MOVE PADDLES
2530 REM 2. SHOW NEW BALL POSITION
2540 REM 3. CHECK FOR COLLISIONS, ETC.
2550 REM
2590 REM ***  FIRST CHECK AND MOVE PADDLES ***
2595 REM
2600 V0=PTRIG(0)-PTRIG(1):IF  NOT V0 THEN 2700
2610 VP0=YP(0)-V0:IF VP0<2 OR VP0>17 THEN 2700
2620 COLOR 0:PLOT 0,YP(0)+V0:COLOR 3:PLOT 0,VP0-V0:YP(0)=VP0
2700 V1=PTRIG(2)-PTRIG(3):IF SINGLE OR V1=0 THEN 3000
2710 VP1=YP(1)-V1:IF VP1<2 OR VP1>17 THEN 3000
2720 COLOR 0:PLOT 39,YP(1)+V1:COLOR 3:PLOT 39,VP1-V1:YP(1)=VP1
3000 REM *** BALL CONTROL ***
3010 COLOR 0:PLOT X,Y
3020 COLOR 1:PLOT XNEW,YNEW
3030 X=XNEW:Y=YNEW
3040 XNEW=XNEW+XMOVE:YNEW=YNEW+YMOVE
3050 IF XNEW<38 AND XNEW>1 THEN 3200
3060 HITP=(XNEW>20):XHIT=39*HITP
3070 IF SINGLE THEN IF HITP THEN 3100
3080 YMSAVE=YMOVE:YNEW=INT(YNEW):YMOVE=(YNEW-YP(HITP))/2
3090 IF ABS(YMOVE)>1 THEN 4000
3100 XMOVE=-XMOVE
3200 IF YNEW=1 OR YNEW=18 THEN YMOVE=-YMOVE
3290 GOTO 2600
4000 REM ***  THE "LOSE" ROUTINE   ******.
4010 COLOR 0:PLOT X,Y
4020 COLOR 1:PLOT XNEW,YNEW
4030 FOR I=1 TO 10:NEXT I
4040 COLOR 0:PLOT XNEW,YNEW
4050 COLOR 2:PLOT XNEW+XMOVE,YNEW+YMSAVE
4130 SOUND 0,132,12,12:POKE 20,0
4140 SETCOLOR 1,0,PEEK(20)*4:IF PEEK(20)<32 THEN 4140
4150 SOUND 0,0,0,0
4200 REM ****   SCORE IT   ********
4210 SCORE(HITP)=SCORE(HITP)+1
4220 LASTWIN=1:IF HITP THEN LASTWIN=-LASTWIN
4990 GOTO 2000
```

```
0000          10          .TITLE "BOING 2 PROGRAM LISTING"
              20 ;
              30 ;FROM COMPUTE! #29 (BILL WILKINSON)
              40 ;
              50 ;BOING2 IN BASIC, INCLUDES PADDLE ROUTINES
              60 ;       IT IS A ONE OR TWO PLAYER GAME
              70 ;I HAVE REMOVED A FEW BUGS IN THE BASIC VERSION -
              80 ;WHICH I CALL : BOING3.BAP
0000          90          .OPT NOEJECT
              0100 ; ASM ,#P:,#D:BOING2.OBJ
              0110 ; LIST #D:BOING2.SRC
```

BOING 2 PROGRAM LISTING
=== GAME STARTUP ===

```
0000          0120          .PAGE "=== GAME STARTUP ==="
              0130 ;
0000          0140          *=    $6000
              0150 BOING
              0160 ;
 :BASIC: 1050 DIM YP(1),SCORE(1):SCORE(0)=0:SCORE(1)=0
6000 4C0760   0170          JMP   AROUND.DIM
6003 00       0180 YP       .BYTE 0,0          ; Y POSITION
6004 00
6005 00       0190 SCORE    .BYTE 0,0          ; AND SCORE
6006 00
              0200 AROUND.DIM
6007 A900     0210          LDA   #0
6009 8D0560   0220          STA   SCORE+0
600C 8D0660   0230          STA   SCORE+1
              0240 ;
 :BASIC: 1050 SINGLE=PEEK(53279)<>7)
600F AD1FD0   0250          LDA   53279        ; PEEK AT CONSOL SWITHCHES
6012 4907     0260          EOR   #$07         ; A=7? THEN A=0. A<>7 : THEN A<>0.
6014 8DE062   0270          STA   SINGLE        ; SET UP OUR FLAG.
              0280 ;
              0290 ;
 :BASIC: 1060 LASTWIN=1: IF RND(0)>=0.5 THEN LASTWIN= -LASTWIN
              0300 ;
6017 A001     0310          LDY   #1           ; USE Y AS TEMP FOR LASTWIN
6019 AD0AD2   0320          LDA   RANDOM       ; GET A RANDOM NUMBE
601C 1002     0330          BPL   HALFCHANCE
601E 88       0340          DEY                ; 50-50 CHANCE THAT WE DO THIS
601F 88       0350          DEY                ; ... MAKES Y = $FF, SAME AS -1
              0360 HALFCHANCE
6020 8CE162   0370          STY   LASTWIN      ; STORE TEMP IN FINAL PLACE
              0380 LINE2000
              0390 ;
              0400 ;
:BASIC: 2010 GR.3 : COLOR 2 : PLOT 0,0 : DRAWTO 39,0
              0410 ;
6023 A903     0420          LDA   #3
6025 20F362   0430          JSR   GRAPHICS
6028 A902     0440          LDA   #2
602A 202063   0450          JSR   COLOR
602D A900     0460          LDA   #0
602F A8       0470          TAY
6030 AA       0480          TAX
6031 202B63   0490          JSR   PLOT         ; PLOT 0,0
              0500 ;
6034 A900     0510          LDA   #0
6036 A227     0520          LDX   #39
6038 A8       0530          TAY
6039 205A63   0540          JSR   DRAWTO       ; DRAWTO 39,0
              0550 ;
              0560 ;
:BASIC: 2020 PLOT 0,19 : DRAWTO 39,19
603C A900     0570          LDA   #0
603E AA       0580          TAX
603F A013     0590          LDY   #19
```

```
6041 202B63 0600          JSR  PLOT       ; PLOT 0,19
6044 A900   0610          LDA  #0
6046 A227   0620          LDX  #39
6048 A013   0630          LDY  #19
604A 205A63 0640          JSR  DRAWTO
            0650 ;
:BASIC: 2030...NOTE WE DON'T PRINT THE SCORES IN THIS VERSION ...
            0660 ;
            0670 ;
            0680 ;
:BASIC: 2040 COLOR 3 : PLOT 0,9 : DRAWTO 0,11 : PLOT 39,9 : DRAWTO 39,11
604D A903   0690          LDA  #3
604F 202063 0700          JSR  COLOR
6052 A900   0710          LDA  #0
6054 AA     0720          TAX
6055 A009   0730          LDY  #9
6057 202B63 0740          JSR  PLOT
605A A900   0750          LDA  #0
605C AA     0760          TAX
605D A00B   0770          LDY  #11
605F 205A63 0780          JSR  DRAWTO     ; DRAWTO 0,11
6062 A900   0790          LDA  #0
6064 A227   0800          LDX  #39
6066 A009   0810          LDY  #9
6068 202B63 0820          JSR  PLOT       ; PLOT 39,9
606B A900   0830          LDA  #0
606D A227   0840          LDX  #39
606F A00B   0850          LDY  #11
6071 205A63 0860          JSR  DRAWTO     ; DRAWTO 39,11
            0870 ;
            0880 ;
:BASIC: 2050 IF SINGLE THEN COLOR 2 : PLOT 39,0 : DRAWTO 39,19
            0890 ;
6074 ADE062 0900          LDA  SINGLE
6077 F016   0910          BEQ  NOTTHEN2050 ; NOT SINGLE PLAYER MODE
6079 A902   0920          LDA  #2
607B 202063 0930          JSR  COLOR      ; COLOR 2
607E A227   0940          LDX  #39
6080 A900   0950          LDA  #0
6082 A8     0960          TAY
6083 202B63 0970          JSR  PLOT       ; PLOT 39,0
6086 A227   0980          LDX  #39
6088 A013   0990          LDY  #19
608A A900   1000          LDA  #0
608C 205A63 1010          JSR  DRAWTO     ; DRAWTO 39,19
            1020 NOTTHEN2050
            1030 ;
            1040 ;
:BASIC: 2060 YP(0)=10 : YP(1)=10
608F A90A   1050          LDA  #10
6091 8D0360 1060          STA  YP
6094 8D0460 1070          STA  YP+1
            1080 ;
:BASIC: 2070 IF SINGLE THEN LASTWIN = 1
6097 ADE062 1090          LDA  SINGLE
609A F005   1100          BEQ  LINE2100   ; NOT SINGLE
609C A901   1110          LDA  #1
609E 8DE162 1120          STA  LASTWIN    ; LASTWIN = 1 (SINGLE <> 0)
            1130 ;
:BASIC: 2100 REM SETUP BALL
            1140 LINE2100
            1150 ;
```

```
                1160 ;
                1170 ;
:BASIC: 2110 XMOVE=LASTWIN:YMOVE=INT(3*RND(0))-1:Y=INT(12*RND(0))+4
60A1 ADE162 1180        LDA  LASTWIN      ; XMOVE = LASTWIN
60A4 8DE362 1190        STA  XMOVE
                1200 ;
60A7 A902   1210        LDA  #2
60A9 208A63 1220        JSR  RND          ; GET RANDOM NUMBER FROM 0 TO 2
60AC 8DE662 1230        STA  Q.YMOVE
60AF CEE662 1240        DEC  Q.YMOVE      ; THEN DO THE '-1'
60B2 0EE662 1250        ASL  Q.YMOVE      ; AND CONVERT TO "HALF-MOVES"
                1260 ;
60B5 A90B   1270        LDA  #11          ; GET RANDOM NUMBER FROM 0 TO 11
60B7 208A63 1280        JSR  RND
60BA 18     1290        CLC
60BB 6904   1300        ADC  #4           ; '+4'
60BD 0A     1310        ASL  A            ; DOUBLE NUMBER OF MOVES TO GET HALF-MOVE
60BE 8DE562 1320        STA  Q.Y          ; AGAIN, THIS IS A "HALF-POSITION" VARIAI
                1330 ;
                1340 ;
                1350 ;
:BASIC: 2120 YNEW=Y : X=19-5*XMOVE : XNEW = X
                1360 ;
60C1 ADE562 1370        LDA  Q.Y
60C4 8DE762 1380        STA  Q.YNEW       ; YNEW = Y
                1390 ; HERE, WE TAKE ADVANTAGE OF THE FACT THAT XMOVE
                1400 ;        CAN ONLY HAVE VALUES -1 OR +1
                1410 ;
60C7 A9FB   1420        LDA  #0-5         ; ASSUME XMOVE = +1
60C9 ACE362 1430        LDY  XMOVE        ; DOES XMOVE = +1 ?
60CC 1002   1440        BPL  XMOVEPLUS    ; YES
60CE A905   1450        LDA  #5           ; NO... SO - 5*-1 = +5
                1460 XMOVEPLUS
60D0 18     1470        CLC
60D1 6913   1480        ADC  #19          ; 19-5 OR 19+5
60D3 8DE262 1490        STA  X
                1500 ;
60D6 ADE262 1510        LDA  X            ; BUT YOU CAN SEE WE DON'T REALLY NEED TH
60D9 8DE462 1520        STA  XNEW         ; XNEW = X
                1530 ;
                1540 ;
:BASIC: 2500 REM *** MAIN PLAYING LOOP ***
                1550 ;
                1560 ;
                1570 ;
:BASIC: 2600 V0=PTRIG(0)-PTRIG(1) : IF NOT V0 THEN 2700
                1580 ; NOTE THAT WHAT WE REALLY WANT IS V0 = +1 IF
                1590 ;        STICK IS PUSHED ONE WAY AND V0 = -1
                1600 ;        IF STICK IS PUSHED THE OTHER.
                1610 ;
                1620 LINE2600
60DC AD7802 1630        LDA  STICK0       ; OS SHADOW LOCATION
60DF 2903   1640        AND  #3           ; LOOK AT JUST FWD AND BACKWD SWITCHES
60E1 4903   1650        EOR  #3           ; INVERT THE SENSE
60E3 F006   1660        BEQ  GOTV0        ; IF ZERO, STICK NOT PUSHED
60E5 C901   1670        CMP  #1           ; FWD PUSHED ?
60E7 F002   1680        BEQ  GOTV0        ; GOOD... WHAT WE WANTED
60E9 A9FF   1690        LDA  #0-1         ; MUST BE PULLED BACK
                1700 GOTV0
60EB 8DEB62 1710        STA  V0           ; TA-DA!
                1720 ;
60EE ADEB62 1730        LDA  V0           ; SO IS STICK PUSHED ?
```

```
60F1 F03E   1740          BEQ  LINE2700  ; IF NOT VO THEN 2700
            1750 ;
            1760 ;
:BASIC: 2610 VP0=YP(0)-V0 : IF VP0 <2 OR VP0>17 THEN 2700
60F3 AD0360 1770          LDA  YP+0
60F6 38     1780          SEC
60F7 EDEB62 1790          SBC  V0
60FA 8DED62 1800          STA  VP0       ; VP0 = YP(0) - V0
            1810 ;
60FD C902   1820          CMP  #2
60FF 9030   1830          BCC  LINE2700  ; IF VP0 <2 THEN 2700
6101 C912   1840          CMP  #18
6103 B02C   1850          BCS  LINE2700  ; OR IF VP0 >17 THEN 2700
            1860 ;
            1870 ;
:BASIC: 2620 COLOR 0:PLOT 0,YP(0)+V0:COLOR 3: PLOT 0,VP0-V0 :YP(0)=VP0
            1880 ;
6105 A900   1890          LDA  #0
6107 202063 1900          JSR  COLOR     ; COLOR 0
            1910 ;
610A AD0360 1920          LDA  YP+0
610D 18     1930          CLC
610E 6DEB62 1940          ADC  V0        ; YP(0)+V0
6111 A8     1950          TAY            ; IS Y POSITION
6112 A900   1960          LDA  #0
6114 AA     1970          TAX
6115 202B63 1980          JSR  PLOT      ; PLOT 0,YP(0)+V0
            1990 ;
6118 A903   2000          LDA  #3
611A 202063 2010          JSR  COLOR     ; COLOR 3
            2020 ;
611D ADED62 2030          LDA  VP0
6120 38     2040          SEC
6121 EDEB62 2050          SBC  V0
6124 A8     2060          TAY
6125 A900   2070          LDA  #0
6127 AA     2080          TAX
6128 202B63 2090          JSR  PLOT      ; PLOT 0,VP0+V0
            2100 ;
612B ADED62 2110          LDA  VP0
612E 8D0360 2120          STA  YP+0      ; YP(0)=VP0
            2130 ;
            2140 ;
:BASIC: 2700 V1=PTRIG(2)-PTRIG(3) : IF SINGLE OR V1=0 THEN 3000
            2150 ;
            2160 LINE2700
            2170 ;NOTE THAT WHAT WE REALLY WANT IS V0 =+1 IF
            2180 ;       STICK IS PUSHED ONE WAY AND -1 IF
            2190 ;       STICK PUSHED THE OTHER.
            2200 ;
6131 AD7902 2210          LDA  STICK1    ; OS SHADOW LOCATION
6134 2903   2220          AND  #3        ; LOOK AT JUST FWD AND BACKWD SWITCHES
6136 4903   2230          EOR  #3        ; INVERT THE SENSE
6138 F006   2240          BEQ  GOTV1     ; IF ZERO, STICK NOT PUSHED
613A C901   2250          CMP  #1        ; FWD PUSHED ?
613C F002   2260          BEQ  GOTV1     ; GOOD...WHAT WE WANTED
613E A9FF   2270          LDA  #0-1      ; MUST BE PULLED BACK
            2280 GOTV1
6140 8DEC62 2290          STA  V1        ; TA-DA!
            2300 ;
6143 ADE062 2310          LDA  SINGLE
6146 D045   2320          BNE  LINE3000  ; IF SINGLE THEN 3000
```

```
6148 ADEC62 2330          LDA  V1          ; SO IS STICK PUSHED ?
614B F040   2340          BEQ  LINE3000    ; OR IF V1=0 THEN 3000
            2350 ;
            2360 ;
:BASIC: 2710 VP1=YP(1)-V1 : IF VP1 < 2 OR VP1 > 17 THEN 3000
            2370 ;
614D AD0460 2380          LDA  YP+1        ; YP(1)
6150 38     2390          SEC
6151 EDEC62 2400          SBC  V1
6154 8DEE62 2410          STA  VP1         ; VP1=YP(1)-V1
            2420 ;
6157 C902   2430          CMP  #2
6159 9032   2440          BCC  LINE3000    ; IF VP1 < 2 THEN 3000
615B C912   2450          CMP  #18
615D B02E   2460          BCS  LINE3000    ; OR IF VP1 > 17 THEN 3000
            2470 ;
            2480 ;
:BASIC: 2720 COLOR 0: PLOT 39,YP(1)+V1 : COLOR 3 : PLOT 39,VP1-V1 : YP(1)=VP1
            2490 ;
615F A900   2500          LDA  #0
6161 202063 2510          JSR  COLOR       ; COLOR 0
            2520 ;
6164 AD0460 2530          LDA  YP+1
6167 18     2540          CLC
6168 6DEC62 2550          ADC  V1          ; YP(1)+V1
616B A8     2560          TAY              ; IS Y POSITION
616C A900   2570          LDA  #0
616E A227   2580          LDX  #39
6170 202B63 2590          JSR  PLOT        ; PLOT 39,YP(1)+V1
            2600 ;
6173 A903   2610          LDA  #3
6175 202063 2620          JSR  COLOR
            2630 ;
6178 ADEE62 2640          LDA  VP1
617B 38     2650          SEC
617C EDEC62 2660          SBC  V1
617F A8     2670          TAY
6180 A900   2680          LDA  #0
6182 A227   2690          LDX  #39
6184 202B63 2700          JSR  PLOT        ; PLOT 39,VP1-V1
            2710 ;
6187 ADEE62 2720          LDA  VP1
618A 8D0460 2730          STA  YP+1        ; YP(1) = VP1
            2740 ;
            2750 ;
```

BOING 2 PROGRAM LISTING
BALL CONTROL

```
618D          2760          .PAGE "BALL CONTROL"
              2770 LINE3000
              2780 ;
              2790 ;
              2800 ;
:BASIC: 3010 COLOR 0 : PLOT X,Y
618D A900     2810          LDA  #0
618F 202063   2820          JSR  COLOR      ; COLOR 0
6192 AEE262   2830          LDX  X
6195 ADE562   2840          LDA  Q.Y
6198 4A       2850          LSR  A          ; DIVEDE HALF-POSITION BY 2 TO GET REAL
6199 A8       2860          TAY
619A A900     2870          LDA  #0
619C 202B63   2880          JSR  PLOT       ; PLOT X,Y
              2890 ;
              2900 ;
 :BASIC: 3020 COLOR 1: PLOT XNEW,YNEW
619F A901     2910          LDA  #1
61A1 202063   2920          JSR  COLOR      ; COLOR 1
              2930 ;
61A4 AEE462   2940          LDX  XNEW
61A7 ADE762   2950          LDA  Q.YNEW
61AA 4A       2960          LSR  A          ; DIVIDE HALF-POSITION BY 2 TO GET REAL
61AB A8       2970          TAY
61AC A900     2980          LDA  #0
61AE 202B63   2990          JSR  PLOT       ; PLOT XNEW,YNEW
              3000 ;
              3010 ;
:BASIC: 3030 X=XNEW : Y=YNEW
61B1 ADE462   3020          LDA  XNEW
61B4 8DE262   3030          STA  X
61B7 ADE762   3040          LDA  Q.YNEW
61BA 8DE562   3050          STA  Q.Y
              3060 ;
              3070 ;
:BASIC:  3040 XNEW=XNEW+XMOVE: YNEW=YNEW+YMOVE
              3080 ;
61BD ADE462   3090          LDA  XNEW
61C0 18       3100          CLC
61C1 6DE362   3110          ADC  XMOVE
61C4 8DE462   3120          STA  XNEW
              3130 ;
61C7 ADE762   3140          LDA  Q.YNEW
61CA 18       3150          CLC
61CB 6DE662   3160          ADC  Q.YMOVE
61CE 8DE762   3170          STA  Q.YNEW
              3180 ;
              3190 ;
              3200 ;
:BASIC: 3050 IF XNEW<38 AND XNEW>1 THEN 3200
              3210 ;
61D1 ADE462   3220          LDA  XNEW
61D4 C926     3230          CMP  #38
61D6 B004     3240          BCS  NOTTHEN3050
61D8 C902     3250          CMP  #2
```

```
61DA B04C   3260         BCS  LINE3200  ; XNEW<38 AND XNEW>1, SO GO
            3270 ;
            3280 NOTTHEN3050
            3290 ;
            3300 ;
:BASIC: 3060 HITP=(XNEW>20):XHIT=39*HITP
            3310 ;
61DC A200   3320         LDX  #0
61DE A000   3330         LDY  #0
61E0 ADE462 3340         LDA  XNEW
61E3 C914   3350         CMP  #20        ; XNEW>20 ?
61E5 9004   3360         BCC  XNEWLT20   ; NO
61E7 A001   3370         LDY  #1         ; YES... SO "TRUE" IS 1
61E9 A227   3380         LDX  #39
            3390 XNEWLT20
61EB 8CE962 3400         STY  HITP
61EE 8EEA62 3410         STX  XHIT
            3420 ;
            3430 ;
:BASIC: 3070 IF SINGLE THEN IF HITP THEN 3100
61F1 ADE062 3440         LDA  SINGLE
61F4 F005   3450         BEQ  LINE3080  ; NOT SINGLE
61F6 ADE962 3460         LDA  HITP
61F9 D024   3470         BNE  LINE3100  ; YES, SINGLE AND HITP
            3480 ;
            3490 ;
:BASIC: 3080 YMSAVE=YMOVE: YNEW=INT(YNEW):YMOVE=(YNEW-YP(HITP))/2
            3500 ;
            3510 LINE3080
61FB ADE662 3520         LDA  Q.YMOVE
61FE 8DE862 3530         STA  Q.YMSAVE  ; YMSAVE=YMOVE
            3540 ;
            3550 ;REMEMBER: WE ARE USING HALF MOVE INCREMENTS
            3560 ;IN Q.Y...VARIABLES...SO WE REALLY SIMPLY
            3570 ;WANT TO GET RID OF THE LOWEST BIT (THE HALF STEP)
            3580 ;
6201 ADE762 3590         LDA  Q.YNEW
6204 29FE   3600         AND  #$FE      ; MASK OFF LAST BIT
6206 8DE762 3610         STA  Q.YNEW    ; YNEW=INT(YNEW)
            3620 ;
6209 AEE962 3630         LDX  HITP      ; SO X IS EITHER 0 OR 1
620C 4A     3640         LSR  A         ; Q.YNEW / 2 GIVES THE TRUE YNEW
620D 38     3650         SEC
620E FD0360 3660         SBC  YP,X      ;YNEW-YP(HITP)
            3670 ;
            3680 ;WE DON'T NEED TO DIVIDE BY 2, BECAUSE Q.YMOVE WANTS HALF-MOVES
            3690 ;
6211 8DE662 3700         STA  Q.YMOVE   ; DONE
            3710 ;
            3720 ;
:BASIC: 3090 IF ABS(YMOVE)>1 THEN 4000
            3730 ;
6214 ADE662 3740         LDA  Q.YMOVE
6217 C903   3750         CMP  #3        ; HALFSTEPS, REMEMBER
6219 9004   3760         BCC  LINE3100  ; 0,1, OR 2 HALFSTEPS
621B C9FE   3770         CMP  #$FE
621D 902C   3780         BCC  LINE4000  ; AHA... >2 HALFSTEPS, <-2 HALFSTEPS
            3790 ;
            3800 ;
:BASIC: 3100 XMOVE= -XMOVE
            3810 LINE3100
621F A900   3820         LDA  #0
```

```
6221 38      3830          SEC
6222 EDE362  3840          SBC  XMOVE
6225 8DE362  3850          STA  XMOVE      ; XMOVE =0-XMOVE
             3860 ;
             3870 ;
:BASIC: 3200 IF YNEW =1 OR YNEW =18 THEN YMOVE=-YMOVE
             3880 LINE3200
6228 ADE762  3890          LDA  Q.YNEW
622B C902    3900          CMP  #1+1       ;REMEMBER : HALF MOVES
622D F004    3910          BEQ  THEN3200
622F C924    3920          CMP  #18+18
6231 D009    3930          BNE  NOTTHEN3200
             3940 ;
             3950 THEN3200
6233 A900    3960          LDA  #0
6235 38      3970          SEC
6236 EDE662  3980          SBC  Q.YMOVE    ; 0-YMOVE
6239 8DE662  3990          STA  Q.YMOVE    ; IS OBVIOUSLY THE SAME AS -YMOVE
             4000 ;
             4010 NOTTHEN3200
             4020 ;
             4030 ;
:BASIC: 3290 GOTO 2600
             4040 ;
             4050 ; IF WE SIMPLY JUMPED BACK TO LINE2600 HERE,
             4060 ; THE GAME WOULD PLAY IMPOSSIBLY FAST...
             4070 ; SO WE PUT IN A DELAY
623C A900    4080          LDA  #0
623E 8D1400  4090          STA  CLOCK.LSB ; THE 60TH OF A SECOND TICKER
             4100 DELAY1
6241 AD1400  4110          LDA  CLOCK.LSB
6244 C902    4120          CMP  #2         ; A 30TH OF A SECOND?
6246 D0F9    4130          BNE  DELAY1
             4140 ;
6248 4CDC60  4150          JMP  LINE2600
             4160 ;
             4170 ;
:BASIC: 4000 REM *** THE LOSE ROUTINE ***
             4180 ;
             4190 LINE4000
             4200 ;
             4210 ;WE WILL SCORE THE MISSES, EVEN THOUGH
             4220 ;WE DON'T DISPLAY THE RESULTS
             4230 ;
             4240 ;
:BASIC: 4010 COLOR 0 : PLOT X,Y
624B A900    4250          LDA  #0
624D 202063  4260          JSR  COLOR
6250 AEE262  4270          LDX  X
6253 ADE562  4280          LDA  Q.Y        ; THE HALF-STEP
6256 4A      4290          LSR  A          ; BECOMES AN INTEGRAL STEP
6257 A8      4300          TAY
6258 A900    4310          LDA  #0
625A 202B63  4320          JSR  PLOT       ; PLOT X,Y
             4330 ;
             4340 ;
:BASIC: 4020 COLOR 1:PLOT XNEW,YNEW
625D A901    4350          LDA  #1
625F 202063  4360          JSR  COLOR
             4370 ;
6262 AEE462  4380          LDX  XNEW
6265 ADE762  4390          LDA  Q.YNEW
```

```
6268 4A      4400          LSR  A          ; AGAIN, HALF STEP TO FULL STEP
6269 A8      4410          TAY
626A A900    4420          LDA  #0
626C 202B63  4430          JSR  PLOT
             4440 ;
             4450 ;
:BASIC:  4030 FOR I=1 TO 10:NEXT I
             4460 ; SHODDY, SHODDY --- USING A FOR/NEXT LOOP FOR TIMING
             4470 ;
             4480 ; HERE WE DO IT RIGHT
626F A900    4490          LDA  #0
6271 8D1400  4500          STA  CLOCK.LSB
             4510 ;
             4520 DELAY2
6274 AD1400  4530          LDA  CLOCK.LSB
6277 C902    4540          CMP  #2          ; TICK-TOCK YET?
6279 D0F9    4550          BNE  DELAY2      ; NOPE, MAYBE JUST TICK
             4560 ;
             4570 ;
:BASIC:  4040 COLOR 0: PLOT XNEW,YNEW
627B A900    4580          LDA  #0
627D 202063  4590          JSR  COLOR
6280 AEE462  4600          LDX  XNEW
6283 ADE762  4610          LDA  Q.YNEW      ; STARTING TO LOOK FAMILIAR?
6286 4A      4620          LSR  A
6287 A8      4630          TAY
6288 A900    4640          LDA  #0
628A 202B63  4650          JSR  PLOT        ; PLOT XNEW,YNEW
             4660 ;
             4670 ;
:BASIC:  4050 COLOR 2: PLOT XNEW+XMOVE,YNEW+YMSAVE
             4680 ;
628D A902    4690          LDA  #2
628F 202063  4700          JSR  COLOR
             4710 ;
6292 ADE462  4720          LDA  XNEW
6295 18      4730          CLC
6296 6DE362  4740          ADC  XMOVE
6299 AA      4750          TAX              ; X REGISTER =XNEW+XMOVE
629A ADE762  4760          LDA  Q.YNEW
629D 18      4770          CLC
629E 6DE862  4780          ADC  Q.YMSAVE
62A1 4A      4790          LSR  A           ; INTEGERIZE THE SUM
62A2 A8      4800          TAY              ; Y REGISTER = YNEW + YMSAVE
62A3 A900    4810          LDA  #0
62A5 202B63  4820          JSR  PLOT        ; PLOT IT.
             4830 ;
             4840 ;
:BASIC:  4130 SOUND 0,132,12,12 : POKE 20,0
62A8 A984    4850          LDA  #132
62AA 8D00D2  4860          STA  SOUND.FREQ ; IMPLICITLY CHANNEL 0
62AD A9CC    4870          LDA  #12*16+12
62AF 8D01D2  4880          STA  SOUND.CONTROL ; 12,12 ALSO FOR CHANNEL 0
62B2 A900    4890          LDA  #0
62B4 8D1400  4900          STA  CLOCK.LSB ; FINALLY, BASIC DID IT RIGHT!
             4910 ;
             4920 ;
:BASIC:  4140 SETCOLOR 1,0,PEEK(20)*4 : IF PEEK(20)<32 THEN 4140
             4930 LINE4140
62B7 AD1400  4940          LDA  CLOCK.LSB ; SAME AS PEEK(20)
62BA 0A      4950          ASL  A
62BB 0A      4960          ASL  A           ; *4
```

```
62BC 8DC502 4970          STA  SETCOLOR1 ; CONTROL REGISTER NUMBER 1
            4980 ;
62BF C980   4990          CMP  #32*4     ; A LITTLE TRICKY. CAN YOU FOLLOW IT?
62C1 90F4   5000          BCC  LINE4140  ; IT WORKS. REALLY
            5010 ;
            5020 ;
:BASIC: 4150 SOUND 0,0,0,0
            5030 ;
62C3 A900   5040          LDA  #0
62C5 8D00D2 5050          STA  SOUND.FREQ
62C8 8D01D2 5060          STA  SOUND.CONTROL
            5070 ;
            5080 ;
:BASIC: 4200 REM *** SCORE IT ***
            5090 ;
            5100 ;
:BASIC: 4210 SCOR(HITP)=SCORE(HITP)+1
62CB AEE962 5110          LDX  HITP
62CE FE0560 5120          INC  SCORE,X   ; ISN'T ASSEMBLER EASY?
            5130 ;
            5140 ;
:BASIC: 4220 LASTWIN=1 : IF HITP THEN LASTWIN = -LASTWIN
            5150 ;
62D1 A901   5160          LDA  #1
62D3 AEE962 5170          LDX  HITP      ; IF HITP?
62D6 F002   5180          BEQ  NOT.HITP  ; NO
62D8 A9FF   5190          LDA  #0-1      ; YES...SO MAKE IT -1
            5200 NOT.HITP
62DA 8DE162 5210          STA  LASTWIN   ; THAT'S ALL THAT IS NEEDED
            5220 ;
            5230 ;
:BASIC: GOTO 2000
            5240 ;
            5250 ;
62DD 4C2360 5260          JMP  LINE2000
            5270 ;
 BOING-- NOT QUITE UP TO PONG
```

BOING 2 PROGRAM LISTING
GENERAL RAM USAGE

```
62E0          5280         .PAGE "GENERAL RAM USAGE"
              5290 ;
62E0 00       5300 SINGLE BRK             ; FLAG FOR ONE-PLAYER GAME
62E1 00       5310 LASTWIN BRK            ; WHO WON LAST TIME?
              5320 ;
              5330 ; THE X MOVES:
62E2 00       5340 X       BRK            ; CURRENT X POSITION
62E3 00       5350 XMOVE   BRK            ; CURRENT X MOVEMENT
62E4 00       5360 XNEW    BRK            ; NEW X POSITION
              5370 ;
              5380 ;AND THE Y POSITIONS AND MOVES:
              5390 ;
              5400 ; REMEMBER: THE Q.YXXX LOCATIONS REFERENCE POSITIONS
              5410 ; OR MOVEMENTS IN TERMS OF HALF STEPS
              5420 ;
62E5 00       5430 Q.Y     BRK            ; CURRENT Y POSITION
62E6 00       5440 Q.YMOVE BRK            ; CURRENT Y MOVEMENT
62E7 00       5450 Q.YNEW BRK             ; CURRENT Y POS'N
62E8 00       5460 Q.YMSAVE BRK           ; SAVED FOR LOSE ROUTINE ONLY
              5470 ;
              5480 ; OTHER MISCELLANY
              5490 ;
62E9 00       5500 HITP    BRK            ; THE HIT PERSON... WHO MISSED.
62EA 00       5510 XHIT    BRK            ; WHERE THE MISS OCCURRED (X POSITION)
              5520 ;
62EB 00       5530 V0      BRK            ; JUST A TEMPORARY
62EC 00       5540 V1      BRK            ; DITTO
              5550 ;
62ED 00       5560 VP0     BRK            ; VERTICAL POSITION OF PADDLE 0
62EE 00       5570 VP1     BRK            ; VERTICAL POSITION OF PADDLE 1
              5580 ;
              5590 ;SYSTEM EQUATES
              5600 ;
0012          5610 CLOCK   =     18        ; THE SYSTEM CLOCK
0014          5620 CLOCK.LSB = CLOCK+2     ; THE 60TH OF A SECOND TICKER
0278          5630 STICK0 =     $278       ; OS SHADOW READ OF FIRST STICK
0279          5640 STICK1 =     $279       ; DITTO FOR 2ND STICK
D200          5650 SOUND.FREQ = $D200      ; PORT WHICH CONTROLS CHANNEL 0 FREQ
D201          5660 SOUND.CONTROL = $D201 ; AND CONTROL
              5670 ;
02C5          5680 SETCOLOR1 = $2C5        ; ALSO KNOWN AS "COLPF1"
62EF          5690         .INCLUDE #D:GRAPHICS.SRC
```

BOING 2 PROGRAM LISTING
GRAPHICS ROUTINES COMPUTE! #21

```
62EF          10          .PAGE "GRAPHICS ROUTINES COMPUTE! #21"
              20 ;FROM COMPUTE #21: GRAPHICS ROUTINES
              30 ;
              40 ; ASM,#P,#D:GRAPHICS.OBJ
              50 ; LIST#D:GRAPHICS.SRC
              60 ; CIO EQUATES
              70 ;
E456          80 CIO      =    $E456     ; CALL OS THRU HERE
0342          90 ICCOM    =    $342      ; COMMAND TO CIO IN IOCB
0344          0100 ICBADR =    $344      ; BUFFER OR FILENAME ADDRESS
0348          0110 ICBLEN =    $348      ; BUFFER LENGTH
034A          0120 ICAUX1 =    $34A      ; AUXILLIARY BYTE #1
034B          0130 ICAUX2 =    $34B      ; AUXIALLARY BYTE #2
0003          0140 COPN   =    3         ; COMMAND OPEN
000C          0150 CCLOSE =    12        ; COMMAND CLOSE
0007          0160 CGBINR =    7         ; COMMAND GET BINARY RECORD
000B          0170 CPBINR =    11        ; COMMAND PUT BINARY RECORD
0011          0180 CDRAW  =    17        ; COMMAND DRAWTO
0012          0190 CFILL  =    18        ; COMMAND FILL (NOT USED IN THIS DEMO.)
0004          0200 OPIN   =    4         ; OPEN FOR INPUT
0008          0210 OPOUT  =    8         ; OPEN FOR OUTPUT
              0220 ;
              0230 ; EQUATES USED BY THE "S:" DRIVER AND
              0240 ;        THE VBLANK ROUTINES
              0250 ;
0055          0260 HORIZONTAL = $55
0054          0270 VERTICAL =  $54
02FB          0280 DRAWCOLOR = $2FB
02C4          0290 COLOR0 =    $2C4
              0300 ;
00FF          0310 LOW    =    $FF
0100          0320 HIGH   =    $100
D20A          0330 RANDOM =    $D20A
              0340 ;
              0350 ;THE ACTUAL ROUTINES
              0360 ;
              0370 ;
              0380 ;=============================================================
```

BOING 2 PROGRAM LISTING
GRAPHICS ROUTINES

```
62EF            0390         .PAGE "GRAPHICS ROUTINES"
                0400 ;
                0410 ;FIRST, SET THE LOCATION AND SOME MISC. RAM USAGE
                0420 ;
62EF 00         0440 SAVECOLOR .BYTE 0 ;WHERE COLOR IS SAVED
62F0 53         0450 SNAME   .BYTE "S:",$9B ;THE FILENAME FOR OPEN
62F1 3A
62F2 00
                0460 ;
                0470 ;
                0480 ;GRAPHICS G
                0490 ;
                0500 ;         ENTRY: A-REG CONTAINS GRAPHICS MODE "G"
                0510 ;         EXIT:  Y-REG HAS COMPLETION STATUS
                0520 ;
                0530 GRAPHICS
62F3 48         0540         PHA             ; SAVE "G"
62F4 A260       0550         LDX  #6*$10     ;FILE 6
62F6 A90C       0560         LDA  #CCLOSE
62F8 9D4203     0570         STA  ICCOM,X
62FB 2056E4     0580         JSR  CIO        ;FIRST, WE MUST CLOSE FILE #6
                0590 ;(WE IGNORE ANY ERRORS FROM THE CLOSE)
                0600 ;
62FE A260       0610         LDX  #6*$10     ; AGAIN, FILE 6
6300 A903       0620         LDA  #COPN      ; WE WILL OPEN THIS "FILE"
6302 9D4203     0630         STA  ICCOM,X
6305 A9F0       0640         LDA  #SNAME&LOW
6307 9D4403     0650         STA  ICBADR,X   ; WE USE THE FILE NAM "S:"
630A A962       0660         LDA  #SNAME/HIGH
630C 9D4503     0670         STA  ICBADR+1,X ; BY POINTING TO IT.
                0680 ;
                0690 ;ALL IS SET UP FOR OPEN, NOW
                0700 ;WE TELL CIO (AND S:) WHAT KIND OF OPEN
                0710 ;
630F 68         0720         PLA             ; OUR SAVED "G" GRAPHICS MODE
6310 9D4B03     0730         STA  ICAUX2,X   ; IS GIVEN TO "S:"
                0740 ;
                0750 ; (NOTE THAT S: IGNORES THE UPPER BITS OF AUX2)
                0760 ;
6313 29F0       0770         AND  #$F0       ; NOW WE GET JUST THE UPPER BITS
6315 4910       0780         EOR  #$10       ; AND FLIP BIT 4
                0790 ;
                0800 ;(READ THE TEXT. S: EXPECTS THIS BIT INVERTED
                0810 ;         FROM WHAT NORMAL BASIC USAGE IS.)
                0820 ;
6317 090C       0830         ORA  #$0C       ; ALLOW READ AND WRITE ACCESS (FOR CIO)
6319 9D4A03     0840         STA  ICAUX1,X   ; MAKE CIO AND S: HAPPY
631C 2056E4     0850         JSR  CIO        ;   AND DO THE OPEN OF S:
631F 60         0860         RTS
                0870 ;=============================================================
                0880 ;
                0890 ;         COLOR C
                0900 ;
                0910 ;
                0920 ;         ENTER: COLOR "C" IN A-REGISTER
```

```
                0930 ;          EXIT: UNCHANGED
                0940 ;
                0950 COLOR
6320 8DEF62     0960            STA  SAVECOLOR     ; EXCITING, WASN'T IT?
6323 60         0970            RTS
                0980 ;
                0990 ;=============================================================
                1000 ;
                1010 ;   POSITION H,V
                1020 ;
                1030 ;        ENTER: H (HORIZ) POSITION IN X,A REGISTERS (LSB,MSB)
                1040 ;               V (VERT) POSITION IN Y-REGISTER
                1050 ;
                1060 ;         EXIT: UNCHANGED
                1070 ;
                1080 POSITION
6324 8655       1090            STX  HORIZONTAL
6326 8556       1100            STA  HORIZONTAL+1 ; READ THE TEXT (MAGAZINE)
6328 8454       1110            STY  VERTICAL  ; TOO SIMPLE, RIGHT?
632A 60         1120            RTS
                1130 ;
                1140 ;=============================================================
                1150 ;
                1160 ; PLOT H,V
                1170 ;
                1180 ;        ENTER: MUST HAVE DONE A PREVIOUS COLOR CALL
                1190 ;               X,A, AND Y REGISTERS SET AS IN POSITION
                1200 ;
                1210 ;
                1220 PLOT
632B 202463     1230            JSR  POSITION
632E A260       1240            LDX  #6*$10    ; FILE 6, AGAIN
6330 A90B       1250            LDA  #CPBINR   ; COMMAND PUT BINARY RECORD
6332 9D4203     1260            STA  ICCOM,X
6335 A900       1270            LDA  #0
6337 9D4803     1280            STA  ICBLEN,X
633A 9D4903     1290            STA  ICBLEN+1,X ; IF BUFFER LENGTH IS ZERO...
633D ADEF62     1300            LDA  SAVECOLOR ; THEN CPBINR PUTS ONE CHAR FROM A-REG
6340 2056E4     1310            JSR  CIO       ; AND THIS IS HOW WE PLOT
6343 60         1320            RTS
                1330 ;
                1340 ;=============================================================
                1350 ;
                1360 ; LOCATE H,V,C
                1370 ;
                1380 ;        ENTER: X,A, AND Y REGISTERS SET UP AS IN POSITION
                1390 ;         EXIT: A-REG HAS THE LOCATED COLOR
                1400 ;               Y-REG HAS THE COMPLETION CODE
                1410 ;
                1420 LOCATE
6344 202463     1430            JSR  POSITION
6347 A260       1440            LDX  #6*$10
6349 A907       1450            LDA  #CGBINR   ; COMMAND GET BINARY RECORD
634B 9D4203     1460            STA  ICCOM,X
634E A900       1470            LDA  #0
6350 9D4803     1480            STA  ICBLEN,X
6353 9D4903     1490            STA  ICBLEN+1,X ; IF BUFFER LENGTH IS ZERO,
6356 2056E4     1500            JSR  CIO       ; THEN THE CHAR IS RETURNED IN A
6359 60         1510            RTS
                1520 ;
                1530 ;=============================================================
                1540 ;
```

```
                1550 ;        DRAWTO H,V
                1560 ;
                1570 ;        ENTER: MUST HAVE DONE A PREVIOUS PLOT
                1580 ;                X,A, AND Y REGISTERS AS IN POSITION
                1590 ;
                1600 ;         EXIT: Y-REGISTER HAS COMPLETION CODE
                1610 ;
                1620 ;
                1630 DRAWTO
635A 202463     1640         JSR  POSITION
635D ADEF62     1650         LDA  SAVECOLOR
6360 8DFB02     1660         STA  DRAWCOLOR    ;WHERE DRAWTO EXPECTS ITS COLOR
6363 A260       1670         LDX  #$10         ;FILE #6
6365 A911       1680         LDA  #CDRAW       ; JUST A COMMAND TO "S:"
6367 9D4203     1690         STA  ICCOM,X
636A A90C       1700         LDA  #$0C
636C 9D4A03     1710         STA  ICAUX1,X     ; INSURANCE
636F A900       1720         LDA  #0
6371 9D4B03     1730         STA  ICAUX2,X     ; ...GUARANTEED TO WORK
6374 2056E4     1740         JSR  CIO          ; DO THE ACTUAL DRAWTO
6377 60         1750         RTS
                1760 ;
                1770 ;=================================================
                1780 ;
                1790 ; SETCOLOR REG,HUE,LUM
                1800 ;
                1810 ;        ENTER: X-REG HAS COLOR REGISTER
                1820 ;               A-REG HAS HUE
                1830 ;               Y-REG HAS LUMINANCE
                1840 ;
                1850 ;         EXIT: (UNDEFINED)
                1860 ;
                1870 SETCOLOR
6378 0A         1880         ASL  A
6379 0A         1890         ASL  A
637A 0A         1900         ASL  A
637B 0A         1910         ASL  A           ; WE NEED HUE * 16
637C 9DC402     1920         STA  COLOR0,X    ; SAVE IT HERE FOR A WHILE
637F 98         1930         TYA
6380 290E       1940         AND  #$0E        ; ONLY LUMINANCE BITS THAT MATTER
6382 18         1950         CLC
6383 7DC402     1960         ADC  COLOR0,X    ; END OF THE WHILE
6386 9DC402     1970         STA  COLOR0,X    ; AND VBLANK WILL MOVE THIS TO HARDWARE
6389 60         1980         RTS
                1990 ;
                2000 ;=================================================
                2010 ;
                2020 ; RND R
                2030 ;
                2040 ;        ENTER: R IN ACCUMULATOR
                2050 ;         EXIT: A-REGISTER CONTAINS A RANDOM
                2060 ;                NUMBER FROM 0 TO R, INCLUSIVE
                2070 ;
                2080 RND
638A 8D9963     2090         STA  RTEMP
638D EE9963     2100         INC  RTEMP       ; MAKES CMP EASIER
                2110 RNDWAIT
6390 AD0AD2     2120         LDA  RANDOM      ; SET A RANDOM NUMBER
6393 CD9963     2130         CMP  RTEMP       ; TOO BIG?
6396 B0F8       2140         BCS  RNDWAIT     ; YES...WAIT
6398 60         2150         RTS
                2160 ;
```

```
6399 00      2170 RTEMP   .BYTE 0          ; RESERVE SPACE
             2180 ;
             2190 ;=============================================
             2200 ;
639A         2210         .END
639A         5700         .END
```

39A = 992 BYTES

BOING 2 PROGRAM LISTING
GRAPHICS ROUTINES

```
 6000 BOING            6007 AROUND.DIM       6003 YP               6005 SCORE
 62E0 SINGLE          =D20A RANDOM           6020 HALFCHANCE       62E1 LASTWIN
 6023 LINE2000         62F3 GRAPHICS         6320 COLOR            632B PLOT
 635A DRAWTO           608F NOTTHEN2050      60A1 LINE2100         62E3 XMOVE
 638A RND              62E6 Q.YMOVE          62E5 Q.Y              62E7 Q.YNEW
 60D0 XMOVEPLUS        62E2 X                62E4 XNEW             60DC LINE2600
=0278 STICK0           60EB GOTV0            62EB V0               6131 LINE2700
 62ED VP0             =0279 STICK1           6140 GOTV1            62EC V1
 618D LINE3000         62EE VP1              61DC NOTTHEN3050      6228 LINE3200
 61EB XNEWLT20         62E9 HITP             62EA XHIT             61FB LINE3080
 621F LINE3100         62E8 Q.YMSAVE         624B LINE4000         6233 THEN3200
 623C NOTTHEN3200     =0014 CLOCK.LSB        6241 DELAY1           6274 DELAY2
=D200 SOUND.FREQ      =D201 SOUND.CONTROL    62B7 LINE4140        =02C5 SETCOLOR1
 62DA NOT.HITP        =0012 CLOCK           =E456 CIO             =0342 ICCOM
=0344 ICBADR          =0348 ICBLEN          =034A ICAUX1          =034B ICAUX2
=0003 COPN            =000C CCLOSE          =0007 CGBINR          =000B CPBINR
=0011 CDRAW           =0012 CFILL           =0004 OPIN            =0008 OPOUT
=0055 HORIZONTAL      =0054 VERTICAL        =02FB DRAWCOLOR       =02C4 COLOR0
=00FF LOW             =0100 HIGH             62EF SAVECOLOR        62F0 SNAME
 6324 POSITION         6344 LOCATE           6378 SETCOLOR         6399 RTEMP
 6390 RNDWAIT
```